A Pruned Fuzzy $k$-Nearest Neighbor Classifier with Application to Electrocardiogram Based Cardiac Arrhythmia Recognition

Fayyaz A. Afsar, M. U. Akram, M. Arif and J. Khurshid

Department of Computer & Information Sciences
Pakistan Institute of Engineering & Applied Sciences (PIEAS)
P.O. Nilore, Islamabad, Pakistan

Abstract— This paper renders a fuzzy nearest neighbor classifier with data pruning to reduce the number of stored prototypes to minimize memory and computational time requirements. The incorporation of fuzzy set theory into nearest neighbor classification makes the decision process more flexible and adaptable to noise in the data. We have also embodied an efficient approach for nearest neighbor search in our algorithm which results in significant reduction in computational time during training and classification. We present results of classification of different data sets from the University of California, Irvine (UCI) machine learning repository to illustrate the effectiveness of the suggested approach for classification purposes. We also give an application of the proposed classification methodology to electrocardiogram (ECG) based recognition of 9 types of arrhythmias using wavelet domain features. The results obtained (~97% accuracy), clearly indicate the effectiveness of this algorithm in the design of a practical ECG analyzer.

Keywords: Pruning, Nearest Neighbor Classification, Fuzzy Logic, Wavelet Transform, Arrhythmia Recognition.

I. INTRODUCTION

The simple $k$-Nearest Neighbor (SKNN) classifier is one of the most commonly used non-parametric approaches to classification problems and it offers many advantages over other classifiers such as simplicity and ease of parallel implementation, adaptability, online learning etc. [1]. The SKNN classifier [1] assigns the class label for a query point through majority voting among its $k$-nearest neighbors. Therefore, the SKNN classifier assigns equal weights to all of the $k$-nearest neighbors regardless of their distances from the query point. An improvement over the SKNN classifier is the Fuzzy $k$-Nearest Neighbor classifier (FKNN) [2] which uses concepts from fuzzy logic to assign a degree of membership of the given query point to different classes while considering the distance of its $k$-nearest neighbors; points closer to the query point contribute a larger value to the membership function of their corresponding class in comparison to far away neighbors. The class with the highest membership function value is taken as the winner. The use of FKNN can be expected to cause reduction in classification error especially in the presence of noise. Moreover it allows for a more flexible decision process.

In certain classification problems, such as ECG based diagnosis of cardiac disorders [3] in which there is large variation across patients and within beats for the same class, the number of training examples required is very large for effective training. While using an instance based classifier, such as $k$-nearest neighbor classification, the training examples are stored as prototypes for subsequent use during classification of an unknown query input. When the size of the training data is large, the amount of storage and time required for performing nearest neighbor search during classification can be a difficult computational challenge. The solution to this problem is to use a pruning algorithm over the training data which would remove some data points from the training dataset without greatly affecting classification accuracy. A variety of pruning algorithms exist in the literature such as Drop-1-5, IB-1-3 etc. A very good introduction to pruning techniques for instance-based learning algorithms is given in [4]. However these pruning approaches consume a large amount of time in pruning which makes their applicability to large data sets difficult.

This paper presents a simple and efficient but effective data pruning algorithm incorporated into fuzzy $k$-nearest neighbor classifier to minimize space and time complexity during classification. Due to its fuzzy nature, this approach has the added advantage of providing the degree of membership of a query beat among different classes. To reduce time complexity further, an efficient nearest neighbor search implementation called ATRIA [5] has been used in the proposed approach. All these features make the classifier presented in this paper, an excellent candidate in the design of a practical classification system such as ECG based cardiac disease diagnosis.

The rest of the paper is organized as follows: Section-II details the proposed Pruned Fuzzy $k$-Nearest Neighbor (PFKNN) classification algorithm. In section-III we analyze the performance of the classifier over different datasets from the UCI machine learning repository [6]. Section-IV gives details about the application of this technique to ECG based classification of 9 types of cardiac rhythms with wavelet
domain feature extraction. Section-V presents the conclusions and future work.

II. PROPOSED ALGORITHM

Consider a training set \( T \) comprising of \( M \) points with the true class label of a point \( x \) being denoted by \( c(x) \). There are two phases of using PFKNN: Training and classification. Training involves pruning of the training data set \( T \) to obtain the prototype set \( P \) and initialization of membership function values of different prototypes in different classes. Classification involves calculation of the membership function values of an unknown query point and assigning its class label. Classification is being explained prior to training as training involves application of the classification procedure.

A. Classification

The approach for classification is given below:

1. Find the \( k \) nearest neighbors \( x_p, j=1...k \) of the given query point in the given prototype training data set \( P \) obtained after training, using fast nearest neighbor search through ATRIA (see figure 1).

2. Evaluate the membership function value of each of the \( N_c \) classes \( (c_i, i=1...N_c) \) using the following relation,

\[
\mu_{c_i}(x) = \frac{\sum_{j=1}^{k} \mu_{c_i}(x_j)d_j^{-2(m-1)}}{\sum_{j=1}^{k}d_j^{-2(m-1)}}
\]

(1)

where \( d_j = \|x-x_j\| \) is the Euclidean distance between \( x \) and \( x_j \). Practically any distance measure can be employed. \( \mu_{c_i}(x_j) \) is the membership value of the point \( x_j \) for class \( c_i \). These membership values are calculated during training. The parameter \( m \) is used to control the effective magnitude of distance of the prototype neighbors from the query point and it can be selected through cross validation along with \( k \). If \( m \) is taken to be infinity then the classifier reduces to a SKNN classifier.

3. The class label of the query point \( x \) chosen by the classification algorithm, \( c_0(x) \), is determined as follows:

\[
c_0(x) = \arg\max_i \left( \mu_{c_i}(x) \right)
\]

(2)

The proposed classifier gives us the membership values of the unknown query beat for different classes allowing a much more informed decision by inclusion of a higher level decision process. To accomplish this, the firing strengths of each class can be normalized and then the following classification rule can be applied: If the normalized firing strength of the runner-up class lies within a certain threshold of that of the winner class then the winning class label can be assumed to be doubtful and the runner-up class label should also be taken into consideration (best-of-the-two classification). Mathematically the winner class label is considered to be doubtful if,

\[
\left( \mu_{\text{winner}}(x) - \mu_{\text{runner-up}}(x) \right) \leq \theta
\]

(3)

This allows us to consider expert opinion when the classifier is not certain about its decision. However, in this paper, we have not utilized this flexibility.

B. Training

Different Steps involved in training are as follows:

1. Start with an empty prototype set, \( P = \phi \).

2. Find \( k \)-Nearest Neighbors, \( x_p, j=1...k \), of each training point, \( x \), such that \( c(x_j) \neq c(x) \) (opposite class neighbors) and add them (without repetition) to the prototype set. This gives us the border points of different clusters in the data (see figure 2).

3. Calculate the membership value of each point, \( x_p \) in the prototype set for each class as follows

\[
\mu_{c_i}(x_p) = \begin{cases} 
0.51 + \frac{0.49k}{k} & \text{if } c(x_p) = c_i \\
0.49 & \text{else}
\end{cases}
\]

(4)

where \( k_i \) is the number of points from the original training set among the \( k \)-nearest neighbors of \( x_p \) that belong to the same class as \( x_p \) itself (same class neighbors).

4. Classify each training point using the prototype set \( P \) and membership function values \( \mu_{c_i}(x_p) \) through the proposed classification procedure detailed earlier (equation 2).

Figure 1 Nearest neighbors of a query point (?) for \( K=3 \)

Figure 2 (a) Opposite class neighbors (filled in black) of a selected point \( x \) and (b) the prototype set elements (filled in black) after step-2 of training

Figure 3 (a) The arrow indicates a training point that will be misclassified given set \( P \) after step 2 of training which is subsequently added to the prototype set (b). (c) The prototype set elements (filled in black) after step-4 of training
If the training point is misclassified, \((c(x_p)≠c(x_n))\) add it to the prototype set and re-evaluate class membership values using (4). This is done in order to accommodate any clusters which may have been missed earlier (see figure 3). The approach given above for data set pruning ensures that no misclassifications occur on the training set. A further enhancement can be to consider a point as a valid prototype point if it has at least a fixed number of nearest neighbors belonging to the same class as the point itself. This can allow effective handling of classification noise.

### III. Empirical Analysis

In this section we present classification results of the application of this algorithm to different real and synthetic datasets. Iris, Glass, Wine and Wisconsin breast cancer (WBCD) datasets have been taken from the UCI Machine Learning Repository [6]. The spiral data set is an artificial dataset and is shown below for illustration. All the datasets were normalized to have zero mean and unity variance along every feature dimension.

![Figure 4 The Spiral Dataset](image)

We have compared the proposed pruning approach with the implementation of Wilson et al. [4] of different pruning algorithms (SKNN, IB2, IB3, DEL and DROP1-5) over different datasets.

In order to have a fair comparison amongst different pruning algorithms we have used 5-fold leave one out cross validation to test these algorithms over different data sets. Presented in table-1 are the mean ± standard deviation values of classification accuracy percentage (A), the percentage ratio (S) of the size of the prototype set retained after pruning to the size of the original training set and the time for classification (T in seconds) over the 5 runs involved in the cross-validation process. All the algorithms were tested on a Dell Optiplex Core 2 Duo CPU operating at 2.66GHz with 1GB of RAM.

This analysis clearly indicates the advantages in terms of computational complexity of using ATRIA for nearest neighbor search as the proposed approach is much faster than other algorithms (~0.6s over the spiral dataset for the proposed approach with ~3s for simple k-NN without ATRIA based nearest neighbor search). The accuracy of the proposed pruning approach is comparable to (or better than) other approaches over these data sets. However a larger (but still manageable) number of points are retained in the prototype set by the proposed algorithm. The amount of time required for pruning large data sets by other pruning methods being considered renders them impractical in case of our application of ECG based diagnosis of cardiac arrhythmias in which the size of the training data is of the order of ~50,000 6-dimensional samples as detailed in the next section.

### IV. Application to ECG Based Arrhythmia Classification

Arrhythmias are irregularities in the pacing of the heart caused by abnormalities in the electrical conduction network or the pacemaker sites in the heart [7]. ECG beat classification, being an integral part of any ECG based automatic decision support system, has been studied by a number of researchers.

| Table 1 This table shows the results of classification (mean ± standard deviation) of different datasets using various pruning algorithms. The proposed method either outperforms or is comparable to other algorithms in terms of classification accuracy. It also indicates that the proposed method using ATRIA based nearest neighbor search is efficient in terms of classification time, especially over large data sets (for convenience the relevant figures have been presented in bold). |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Dataset | Proposed | SKNN | IB2 | IB3 | DEL | DROP1 | DROP2 | DROP3 | DROP4 | DROP5 |
| Iris | A | 95.3±7.06 | 94.29±5.64 | 87.14±10.54 | 94.29±5.67 | 86.57±10.21 | 87.86±8.28 | 93.57±7.10 | 93.57±8.55 | 93.57±8.55 | 92.14±7.10 |
| N=150 | S | 16.22±0.95 | 100±0 | 15.37±2.26 | 20.66±3.72 | 9.26±1.39 | 12.35±1.5 | 16.84±1.17 | 15.22±1.59 | 15.22±1.59 | 12.5±1.39 |
| Glass | A | 73.49±1.27 | 72.86±3.98 | 68.57±6.16 | 67.62±7.64 | 70.48±7.26 | 62.38±7.02 | 67.62±4.94 | 65.24±4.94 | 69.52±7.02 | 63.81±6.61 |
| N=9 | S | 75.91±1.05 | 100±0 | 42.21±2.2 | 30.7±3.87 | 39.77±3.2 | 25.5±9.2 | 23.26±4.9 | 24.88±2.71 | 28.84±2.62 | 26.28±1.04 |
| WBCD | A | 93.51±3.08 | 96.64±2.29 | 93.28±3.1 | 94.34±1.01 | 90.44±3.6 | 92.39±3.68 | 94.51±2.38 | 95.58±2.5 | 95.58±2.17 | 95.4±1.92 |
| N=150 | S | 29.49±1.45 | 100±0 | 67.3±3.12 | 8.55±1.49 | 3.64±0.51 | 5.44±1.01 | 11.4±1.36 | 8.86±0.99 | 9.61±3.15 | 7.06±0.91 |
| Wine | A | 0.03±0.01 | 0.01±0.01 | 0.01±0.01 | 0.01±0.01 | 0.04±0.01 | 0.03±0.01 | 0.03±0.01 | 0.03±0.01 | 0.03±0.01 | 0.03±0.01 |
| N=13 | S | 0.05±0.01 | 0.02±0.01 | 0.03±0.01 | 0.03±0.01 | 0.07±0.01 | 0.08±0.01 | 0.06±0.01 | 0.06±0.01 | 0.06±0.01 | 0.07±0.01 |
| Spiral | A | 96.55±0.49 | 97.06±0.38 | 93.89±0.5 | 94.2±0.39 | 92.64±1.40 | 91.0±1.07 | 96.47±0.44 | 96.51±0.26 | 96.57±0.24 | 96.42±0.61 |
| N=5280 | S | 19.21±0.25 | 100±0 | 8.3±0.35 | 7.18±0.67 | 5.27±0.15 | 6.65±0.39 | 11.6±0.37 | 10.78±0.29 | 11.04±0.29 | 7.45±0.38 |
| T | 0.6±0.03 | 3.06±0.01 | 2.77±0.02 | 3.24±0.08 | 19.5±0.09 | 12.6±0.08 | 16.28±0.07 | 16.5±0.1 | 16.5±0.22 | 18.0±0.23 |
Different feature extraction methods for beat classification available in the literature include use of Fourier Transform [8], multi-resolution analysis [9], wavelet transform [10-13], independent component analysis [14], morphological analysis [15, 16] etc. For the purpose of classification of the extracted features, the literature reports a variety of classifiers such as Backpropagation Neural Networks [17], Learning Vector Quantization and Probabilistic Neural Networks [10], Fuzzy Inference Systems [18], Nearest Neighbor classifiers [15, 16] etc.

In our previous work [13] we have used features extracted from two-level wavelet decomposition of an ECG signal. The wavelet decomposition was performed through algorithm a’trous using the wavelet proposed by Martínez et al. [19] for effective and accurate QRS delineation, which is an important pre-processing step for beat classification in which the onset, offset and fiducial point of the QRS complex are detected. This wavelet offers inherent noise suppression and eliminates the need for re-evaluation of wavelet coefficients for beat classification as these are already obtained during QRS detection and delineation. A simple $k$-nearest neighbor (SKNN) classifier had been employed in our previous work for the classification of 6 types of beats (Paced Beats (PB), Atrial Premature Beat (APB), Premature Ventricular Contraction (PVC), Normal (N), Left and Right Bundle Branch Blocks (LBBB & RBBB)) to give an accuracy of ~99.5% over selected records (11,600 beats for training & testing each) from the MIT-BIH Arrhythmia database [20] with high noise tolerance and robustness against decrease in the size of the training data set. However with increasing number of beats in the training set, the use of SKNN classifier becomes infeasible due the large memory requirements and the amount of time for nearest neighbor search. Therefore we have applied the proposed classification methodology with pruning to reduce these computational requirements while maintaining high accuracy in beat classification.

In this section, we present the feature extraction and classification stages involved in the design of a practical ECG based beat classification system for the classification of 9 types of cardiac rhythms: Ventricular Fibrillation (VF), Paced Beats (PB), Atrial Premature Beat (APB), Fusion of Normal and Ventricular Beats (FNV), Left and Right Bundle Branch Blocks (LBBB & RBBB), Normal (N), Premature Ventricular Contraction (PVC) and Fusion of Normal & Paced beats (FPN). Major system components include:

- Feature Extraction
- Normalization
- Dimensionality Reduction
- Classification through PFKNN

The details of each of these components are given below:

### A. Feature Extraction

For feature extraction we have used the same wavelet as in [19] with wavelet transform implemented through Algorithm a’ trous. The wavelet is taken to be the derivative of a low pass filter which offers inherent noise suppression. This wavelet is given by,

$$
\Psi(\Omega) = j\Omega \left( \frac{\sin\left(\frac{\Omega}{4}\right)}{\frac{\Omega}{4}} \right)^4 \quad (5)
$$

From the implementation viewpoint, it can be implemented through FIR low pass ($H$) & high pass ($G$) filters whose frequency responses are given by:

$$
H(e^{j\omega}) = e^{\jmath \omega} \left( \cos\frac{\omega}{2} \right), \quad G(e^{j\omega}) = 4 je^{\jmath \omega} \left( \sin\frac{\omega}{2} \right) \quad (6)
$$

The same wavelet transform can be used for detection and delineation of the QRS complex. For the purpose of beat classification, we utilize wavelet coefficients of a 64 point window centered at the QRS fiducial point up to scale $2^4$ (see figure 5).

![Figure 5](image)

**Figure 5** The wavelet decomposition for a representative QRS Complex [13]

The following 11 features are extracted from the ECG signal and the corresponding wavelet decomposition coefficients:

i. Variance of the original QRS complex signal denoted by $\sigma_S^2$

ii. Variance in each sub-band denoted by $\sigma_{A2}^2, \sigma_{D2}^2, \sigma_{D3}^2$

iii. Variance of the autocorrelation function of wavelet coefficients in each sub-band denoted by $\sigma_{R(A2)}^2, \sigma_{R(D2)}^2, \sigma_{R(D3)}^2$

iv. Ratio of minimum to maximum wavelet coefficient in each sub-band denoted by $r_{A2}, r_{D2}, r_{D3}$

These features are combined with the instantaneous RR interval to produce a feature set given by \( \{\sigma_S^2, \sigma_{D1}^2, \sigma_{R(D1)}^2, r_{D1}, \sigma_{D2}^2, \sigma_{R(D2)}^2, r_{D2}, \sigma_{A2}^2, \sigma_{R(A2)}^2, r_{A2}, RR\} \) for a single beat.

### B. Normalization

A normalization process is necessary to standardize all features to the same level. Tangent sigmoid function is used for the normalization as given below,

$$
x_{\tilde{x}} = \text{tansig} \left( \frac{x - \bar{x}}{\sigma_x} \right) \quad (7)
$$
where $\bar{x}_j$ and $\sigma_j$ are the mean and the variance of the $j^{th}$ component of the feature vector across all the training beats. This function will normalize the range of features to [1, 1]. The normalized feature set for the $k^{th}$ beat is denoted by $F_k^j$.

C. Dimensionality Reduction

The extracted features except the RR interval are subjected to Principal Component Analysis [21]. RR interval is treated separately because of its temporal nature. A covariance matrix is formed on the basis of the first ten features and its eigen values and vectors are computed. Five of the 10 eigen vectors corresponding to the highest eigen values are retained as they capture about 98% of energy in the features [13]. The input data is then projected onto these selected bases and normalized RR interval values are appended to the projected feature set to result in a 6 dimensional feature space.

D. Classification using PFKNN

Classification of an unknown query beat (with normalized features) is done through the proposed PFKNN classifier explained earlier after training.

V. RESULTS & DISCUSSION

In this section we present the results of classification of the 9 types of cardiac rhythms discussed earlier through the proposed methodology. These results are given in terms of the following performance metrics:

a. Mean & Standard Deviation of Positive Predictive Values (PPV) of each class over 5 runs with random selection of training & testing data points. With $TP_c$ and $FP_c$ representing the number of true and false positives respectively for a given class $c$, its PPV is defined by,

$$PPV_c = \frac{TP_c}{TP_c + FP_c} \times 100 \quad (8)$$

b. Mean & Standard Deviation of Sensitivity Values ($Se_c$) of each class over 5 runs with random selection of training & testing data. If $FNC_c$ is the number of false negatives for a class $c$, its Sensitivity is defined by,

$$Se_c = \frac{TP_c}{TP_c + FNC_c} \times 100 \quad (9)$$

c. Mean & Standard Deviation of Total Accuracy ($A$) of each class over 5 runs with random selection of training & testing data. Total Accuracy is define by,

$$A = \left(1 - \frac{N_{error}}{N_{test}}\right) \times 100 \quad (10)$$

where $N_{error}$ is the number of misclassifications and $N_{test}$ is the total number of testing beats for all classes.

Another performance measure, for analyzing the performance of pruning algorithms is the percentage ratio (S) of the size of the prototype set retained after pruning to the size of the original training set.

The system has been evaluated using data from the MIT-BIH Arrhythmia Database corresponding to the lead ML-II with a total of ~104,700 beats. The distribution of these beats into different classes is as follows:

Table 2 Number of Beats of Each class included in the study

<table>
<thead>
<tr>
<th>Class</th>
<th>SE</th>
<th>PPV</th>
</tr>
</thead>
<tbody>
<tr>
<td>VF</td>
<td>81.02±2.44</td>
<td>79.12±1.94</td>
</tr>
<tr>
<td>PB</td>
<td>99.55±0.15</td>
<td>99.73±0.13</td>
</tr>
<tr>
<td>APB</td>
<td>83.79±0.83</td>
<td>79.11±0.96</td>
</tr>
<tr>
<td>FNV</td>
<td>75.17±2.86</td>
<td>72.64±2.02</td>
</tr>
<tr>
<td>LBBB</td>
<td>94.48±0.33</td>
<td>93.61±0.44</td>
</tr>
<tr>
<td>N</td>
<td>97.96±0.07</td>
<td>98.34±0.04</td>
</tr>
<tr>
<td>RBBB</td>
<td>98.48±0.13</td>
<td>97.80±0.21</td>
</tr>
<tr>
<td>PVC</td>
<td>92.13±0.38</td>
<td>92.18±0.67</td>
</tr>
<tr>
<td>FPN</td>
<td>76.58±2.81</td>
<td>83.09±3.92</td>
</tr>
<tr>
<td>Accuracy (A)</td>
<td>96.75±0.07</td>
<td></td>
</tr>
<tr>
<td>S</td>
<td>21.04</td>
<td></td>
</tr>
</tbody>
</table>

This shows that only about 21% points are retained in the prototype set while offering a high classification accuracy of ~97%. The time taken for this classification is 3.53s with 41.4s being spent on pruning. The classification time taken by SKNN without ATRIA is ~1116.76s on the same machine. This reduction in classification time (by a factor of ~300), very clearly illustrates the effectiveness of the incorporation of efficient nearest neighbor search using ATRIA.

The table below compares different approaches for beat classification available in the literature with the proposed methodology. However a fair comparison is not possible amongst these approaches as different authors have used different number of beat types, different number of beats for training & testing and different data sets. This table shows that the proposed approach compares well with other methods in the literature.

Table 4 Comparison of different beat classification methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Number of Beat Types</th>
<th>Number of Features</th>
<th>Number of Beat</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oowski et al. [22]</td>
<td>7</td>
<td>18</td>
<td>7,185</td>
<td>96.06</td>
</tr>
<tr>
<td>Dokur et al. [23]</td>
<td>5</td>
<td>15</td>
<td>1,000</td>
<td>97.00</td>
</tr>
<tr>
<td>Prasad et al. [9]</td>
<td>12</td>
<td>25</td>
<td>105,423</td>
<td>96.77</td>
</tr>
<tr>
<td>Yu et al. [12]</td>
<td>6</td>
<td>11</td>
<td>23,200</td>
<td>99.65</td>
</tr>
<tr>
<td>Yu et al. [24]</td>
<td>8</td>
<td>17</td>
<td>9,800</td>
<td>98.7</td>
</tr>
<tr>
<td>Proposed</td>
<td>9</td>
<td>6</td>
<td>104,700</td>
<td>96.75±0.07</td>
</tr>
</tbody>
</table>
VI. CONCLUSIONS & FUTURE WORK

In this paper we have proposed an efficient approach for classification of 9 types of cardiac arrhythmias through ECG using wavelet domain features with PCA. The proposed classification methodology utilizes data pruning and efficient nearest neighbor search in order to reduce classification time. Moreover the results of the application of the classification procedure on classical machine learning data sets has also been presented in order to further clarify the advantages brought in through the PFKNN classifier. In future we would like to analyze, mathematically, the proposed approach in terms of computational and memory complexity. In reference to ECG based beat classification, a look at the sensitivities and positive predictive values of different classes in table-2, shows that the PPV and Se values for classes with fewer numbers of beats in the data are low. This results from the imbalanced-dataset problem because the number of beats of some of the classes (see table 2) are very small in comparison to other classes. In future, we would like to extend the proposed methodology to handle the data imbalanced data set problem.

REFERENCES