A Novel Neighbor Selection Approach for KNN: A Physiological Status Prediction Case Study

Fatemeh Kaveh-Yazdy
School of Mathematics,
Computer Science Department,
Yazd University
Yazd, Iran. P.O Box 89195-741
Kavehy_f@yahoo.com,

Mohammad-Reza Zare-Mirakabad
School of Electrical and Computer Engineering,
Yazd University,
Yazd, Iran. P.O Box 89195-741
mzare@yazduni.ac.ir

Feng Xia
School of Software,
Dalian University of Technology,
Dalian 116620, China
f.xia@ieee.org

ABSTRACT
Conventional weighting KNNs enhance the accuracy of label selection by weighting the neighbors. In this paper, however, we propose a novel weighting approach which weights the distances to find the neighbors more accurately. We take importance of size of classes and dispersity of samples into account for this purpose. Moreover we use LDA that saves discrimination level of data for reducing dimension of problem space. We show the effectivity of our proposed method on PDMC-04 dataset to predict physiological status of human subjects.

Categories and Subject Descriptors
H. 2.8. **[Database Applications]**: Data mining– Classification, Prediction.
J. 3. **[Life and Medical Sciences]**: Health, Medical Information– physiological status prediction.

General Terms
Algorithms

Keywords
Physiological status; Weighted K Nearest Neighbor; Class Dispersity; Linear Discriminant Analysis.

1. INTRODUCTION
Nowadays, sensor-based monitoring has become a ubiquitous part of modern life. Wide ranges of sensor usages, from surveillance applications to healthcare monitoring lead to tremendous amounts of data and very large databases. Collected data of sensor networks can be used in modeling environments, systems or events, and predicting their future behavior. Sensor arrays generate sequential and large size flows of data that are tend to be affected by several noise resources and should be cleaned before using in any other processing step. Sensed data as output of these networks can be used during modeling of dynamic systems and predicting their future behavior. Body sensor network as a large family of sensor networks are employed to sense several kinds of physiological data like heart rate, blood pressure, oxygen saturation and body temperature. Two main reasons that encourage modern health-care systems to use BSNs are: (a) to enhance the quality of in-site medical services and (b) to enhance the quality of life for in-home healthcare services [21].

ContextDD'12, August 12, 2012, Beijing, China.

Sensed data of BSNs are time series of vital signs and physiological bio-makers that can be used to monitor human body. After collecting data of each patient, clinicians can personalize clinical/home care [22]. Human body, from a mathematical point of view, can be defined as a system and its physiological behavior can be sensed and monitored in order to predict the future status. Physical status prediction and task of medical prognosis are related tasks, which use short-term trajectory of physiological key indicators. Wearable sensor networks [16] and implantable biosensors [1] are employed in automotive pervasive monitoring of patients and/or for detecting abnormal patterns of vital signs. Life-threatening abnormalities fire medical alarms and notify physicians to decrease the response time [15]. All these tasks are employed in clinical decision support systems. Osherof et al. [18] define clinical decision support system as software that "Provide clinicians, patients or individuals with knowledge and person-specific or population information, intelligently filtered or presented at appropriate times, to foster better health processes, better individual patient care, and better population health".

Sleep apnea detection [5], prediction of acute hypotensive episodes [8], fluid estimation [6] are samples of predicting a physiological status based on sensor data in hospitals and healthcare centers. Models of status prediction are basis of medical decision support systems [12], intelligent monitoring [17, 22] and smart alarming [13]. Ever increasing requests for clinical decision support systems and requirement of real-time automated data analysis for very large in-hospital databases yield data mining as an ideal candidate tool. Data mining, pattern recognition and machine learning are tied up together. Data mining adopt conventional methods of pattern recognition and machine learning. Classification and clustering are popular pattern recognition tools whether the data are structured (like numerical, discrete and labels of bins) or unstructured (like descriptive textual data of health records). Rokach and Maimon [19] categorize classification methods as a child of prediction methods in taxonomy of data mining methods. Classification might be defined as the task of assigning sample vectors to one of members of finite discrete set of categories (called classes). K nearest neighbor, a typical lazy learner method, assigns class labels of new records using common sense analogy, which is provided by cross comparing of examples and all of their neighbors and polling over labels of their nearest neighbors. There are many extensions for kNN such as weighting kNN, kernel based kNN [24], and density based versions [20]. Our original idea comes back to change conventional weighting approaches. In typical weighting extensions of kNN, weights are applied on k nearest neighbors to enhance the accuracy of label

selection; we, however, weight the candidates before neighbor selection to move up some of neighbors might not be selected without weighting. We exactly aim at this problem by taking into account the in-class distributions of samples that enhances accuracy by applying weights before selection of neighbors.

The rest of this paper is organized as follows: Section 2 introduces the background behind our proposed weighting approach and our motivation for proposing it. In section 3 we introduce our data representation framework. In section 4 we describe our selected methodology. Section 5 includes dataset description and a precise comment on experiment framework as well as it shows the results of experiments. Finally, we present our findings on experiments in section 6 to highlight conclusions and discuss the results of comparisons.

2. MOTIVATION

Typical k nearest neighbor algorithms use distance measures to find nearest neighbors of input data point and to classify it according to their labels. Weighting-based kNN algorithms use different strategies such as distance based weighting [7, 10] and feature ranking [23] to consider the influence of features and neighbors for labeling input data points. To the best of our knowledge, what typical weighting methods do not consider is distribution of neighbors in classes, while data points have been dispersed in classes, especially when scatterness of samples near class boundaries is high. Consequently, it is possible that a data point which semantically belongs to i*th* class falls in j*th* class, only because it is nearer to the j*th* class. This is our motivation to propose effectivity-based weighting approach which takes into account the effectivity of membership of a data point into a class even though it is nearer to another class.

Larger classes may have more suspicious points in their neighbors, and the probability of a data point (in boundary area) as a member of a large class, is much more than its probability in a small class, thus a neighbor from large class should have more influence on labeling mechanism. Suppose that we have two classes (say C_1 and C_2) and the distance between input data point and a neighbor from class C_1 is d_1 and its distance from a neighbor of C_2 is d_2. Let $d_1<d_2$ and C_1 is smaller than C_2. What we are going to do is proposing a weighting approach using effectivity measures so that $w_2d_2<w_1d_1$. That is, we propose a weighting mechanism that adjusts distances regarding to class importance. In other words, effectivity measures balance the distances according to importance of neighbors and shrink the small classes. The proposed weighting mechanism expresses the effectivity of a neighbor in a class and the effectivity of the class in the problem domain.

3. DATA REPRESENTATION

Following we introduce notations and definitions as well as mathematical formulation of computing interestingness measures used in our proposed effectivity based weighting kNN.

Let X_i, ($X_i = \langle x_1, x_2, ..., x_p \rangle$), denotes each record of training set as input data that can be shown by p-dimensional feature vector and has an attached-element L_i as annotated label. Labels are selected from a *q*-size set of classes. While features in training set are real continuous values, the discretization preprocess can be applied to reduces numerosity of values and simplifies computations for continuous variables. Discretized version of training set formed in the same way as defined for training set with same set of labels except that elements of this version are positive integer values. Large amount of records that are expressed in small set of discrete values include many repeated patterns. In order to categorize training set, all repeated patterns should be extracted and recorded in a category set to express diversity of patterns in dataset and classes.

Definition 1 (Category set). *Category set is defined by keeping only one instance for a group of repeated patterns, so its records have same features as training set records but g records where g << n, i.e.*

$$\text{Category set} = \begin{bmatrix} \overbrace{\phantom{x_{11}\ x_{12}\ \cdots\ x_{1p}}}^{A_1 \quad A_2 \quad \cdots \quad A_p} \\ x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{g1} & x_{g2} & \cdots & x_{gp} \end{bmatrix}$$

Following SQL command shows categorization procedure.

```
Select *
From    Discretized_Training_set
Group by  x1, x2, x3,…, xp
```

We then compute frequency of each pattern in whole training set as well as frequency of appearance in each class of labels.

Definition 2 (A-frequency matrix). *Count of all instances of a category C_i appeared in a specified class j is defined as in-class A-frequency (Appearance frequency), f_{ij}. Consequently, count of all instances of each extracted category in whole training set is defined as total A-frequency F_i, i.e.*

$$F_i = \sum_{j=1}^{q} f_{ij} \tag{1}$$

A-frequency matrix includes total A-frequency and in-class A-frequencies for all categories separately, i.e.

$$A - Frequency = \left[\begin{array}{c|cccc} \overbrace{F_1}^{Total} & \overbrace{f_{11}}^{Class_1} & \overbrace{f_{12}}^{Class_2} & \cdots & \overbrace{f_{1q}}^{Class\ q} \\ F_2 & f_{21} & f_{22} & \cdots & f_{2q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ F_g & f_{g1} & f_{g2} & \cdots & f_{gp} \end{array}\right]$$

Following SQL commands show the mechanism allowing us to extract patterns and compute their total A-frequency, and in-class A-frequencies respectively.

```
Select  *, Count (*)
From    Discretized_Training_set
Group by  x1, x2, x3,…, xp
```

```
Select  *,count(*)
From    Discretized_Training_set
Group by  x1, x2, x3,…, xp
Where  class="i"
```

Imbalance distribution of patterns in classes leads to zero in-class frequency values for some of patterns in category set. To avoid

division-by-zero during computations we use Laplace smoothing to avoid these cases in calculations [14].

Definition 3 (Appearance Probability Matrix). *According to A-frequency values we compute appearance probabilities of patterns. In this way, for each pattern record, we compute its probability of it in all class labels separately i.e.*

$$pr_{ij} = \frac{f_{ij}}{n} \tag{2}$$

These values indicate probability that each record belongs to each category and its corresponding class label. The total A-probability is computed from the total A-frequency of each pattern divided by number of records i.e.

$$\Pr_i = \frac{\sum_{j=1}^{g} f_{ij}}{n} = \frac{Tf_i}{n} \tag{3}$$

A-probability matrix includes total A-probability and in-class probability values for all of categories, similar to A-frequency matrix.

$$A-Probability = \left[\begin{array}{c|ccccc} \overbrace{Total}^{} & \overbrace{Class_1 \;\; Class_2 \;\; \cdots \;\; Class\ q}^{} & & & & \\ \Pr_1 & pr_{11} & pr_{12} & \cdots & pr_{1q} \\ \Pr_2 & pr_{21} & pr_{22} & \cdots & pr_{2q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \Pr_g & pr_{g1} & pr_{g2} & \cdots & pr_{gp} \end{array}\right]$$

Definition 4 (Effectivity Measure). *Different effectivity measures are defined based on described variables and u. let u=1/q be the average probability for all (uniformly distributed) classes. Each effectivity measure is a computed column of probability-based interestingness measures for diversity of patterns in class labels, like:*

$$E_{<\text{E-measure's name}>} = \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_g \end{bmatrix}$$

Seven different effectivity measures are defined similarly to [11]. These measures are employed during classic data mining methods as interestingness measures but we utilized them in the new style of using. Accordingly, for each effectivity measure we generate E vectors, filled with values that are computed by formula shown in Table 1. KNN algorithms are lazy learners that postpone the computations until classification of test set. All formulated processes as preprocessing phase for the training set needs to be done on the test set. Test set data are continues too, so we define discretized version of test set with same method that is used for discretization of training set. Same as typical kNN algorithms we compute distance between pairs of samples, and then we weighted them according to our weighting approach to select nearest neighbors.

Definition 5 (Weighted distance). *Let y_i be a sample selected from test set and x_j is an instance record from training set. Let d_{ij} be the pure distance between y_i and x_j.*

$$d_{ij} = dist\left(y_i, x_j\right) \tag{4}$$

For each x_j exists a pattern C_t in category set that is same as x_j.

$$\exists\, C_t \in Category\ \ set \Rightarrow x_j = C_t \tag{5}$$

All distance measures are weighted by effectivity measures. Each pure distance d_{ij} multiplied by the effectivity measure of category of its neighbor x_j. x_j belongs to category C_t, thus e_t shows the interestingness of this neighbor throughout the domain of problem.

$$D_{ij} = d_{ij} \times e_t \tag{6}$$

Accordingly, weighted distance matrix includes distances between all combinations of samples from training and test sets.

$$Weighted\ \ Dist.\ Matrix = \begin{bmatrix} D_{11} & D_{12} & \cdots & D_{1n} \\ D_{21} & D_{22} & \cdots & D_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ D_{m1} & D_{m2} & \cdots & D_{mn} \end{bmatrix}$$

For searching nearest neighbors, we use weighted distances and choose k neighbors with minimum weighted-distance to test record. Weighting approach changes the sequence of distances and lets some of neighbors to move up in array of neighbors and push some of them down, simultaneously.

It might be worth reminding the reader that our algorithm uses weighted distances for neighbor selection and do not postpone weighting until labeling step like most of typical weighting approaches.

Table 1. Interestingness measures used as effectivity measures

Title	Formula
Shannon	$e_i = \sum_{j=1}^{q} pr_{ij} \log_2 pr_{ij}$
Gini	$e_i = \frac{q}{2} \sum_{j=1}^{q} \sum_{k=1}^{q} \lvert pr_{ij} - pr_{ik} \rvert$
Lorenz	$e_i = q \sum_{j=1}^{q} (q - i + 1)\ pr_{ij}$
Schutz	$e_i = \frac{1}{2qu} \sum_{j=1}^{q} \lvert pr_{ij} - u \rvert$
Atkinson	$e_i = 1 - \prod_{j=1}^{q} \frac{pr_{ij}}{u}$
Whittaker	$e_i = 1 - \frac{1}{2} \sum_{j=1}^{q} \lvert pr_{ij} - u \rvert$
Kullback	$e_i = \log_2^q - \sum_{j=1}^{q} pr_{ij} \log_2 \frac{pr_{ij}}{u}$

4. METHODOLOGY

We employ two phases to predict the physiological status of each record of test set. In the first phase, a feature extraction pre-process applied to reduce dimension of data set and the second phase utilize our proposed weighted k-nearest neighbor classifier to predict label of records. Development steps of phases are described in details in next two subsections.

4.1 Phase I: Cleaning and Dimension Reduction

Feature selection methods reduce size of data by eliminating redundant dimensions of data. Most of these methods try to select a representative column for a subset of correlated columns. Facts, events, objects and human beings are observed by their attributes. More features may model a target fact accurately or increase the

computational complexity without any improvements. Data preprocessing step started with cleaning all rows with abnormal patterns. Same process were done for test set data records, to clean abnormal patterns, normalize values and quantize with same specifications as like as training set. Linear Discriminant Analysis (LDA) is one of the most important methods from linear dimension reduction methods. LDA discriminates data and saves between class distances. Applying LDA on the training data and transforming the feature space into a new subspace drop the advantages of discrete data, so we use LDA with SRDA [9] implementation in order to generate eigenvalues of feature matrix and feature selection. Top nonzero eigenvalues and their corresponding columns are selected for next step of our algorithm. Choosing columns without transforming into new space saves the delicate nature of discretization.

4.2 Phase II: Weighted kNN

Min-max is employed to transform linearly float values of features into [0,1] range. Normalized data are quantized uniformly to B bins. For each pattern total A-frequency and in-class A-frequencies are computed and smoothed to avoid zero A-frequencies. A-Probability matrix of patterns which is calculated from A-frequency matrix is used in computing the interestingness measures of patterns. Distance measures used in this paper are Euclidean, complexity-invariant distance measure for time-series [4] and edit distance. Using three distance measures leads to compute three matrices of pure distances between all samples of test set and records of training set. For each pair of data points, we track the vector of training set sample in list of patterns and use the effectivity measure of this sample's category to adjust its pure distance. Eight different interestingness measures, from a wide area of applications were selected.

To consider the significance of larger classes, whenever it is convenient, in cases which interestingness has inverse correlation with size, we use inverse value of interestingness measures as effectivity. Accordingly, total A-probability, Gini, Schutz, Atkinson and Lorenz are inversed and multiplied by pure distances. Shannon, Whittaker and Kullback measures are used in normal form. For each sample of test set we choose k nearest neighbors according to the proposed weighted distance approach. It was pointed out earlier that the weights in our algorithm are applied before neighbor selection phase of kNN. Label of each test set sample is determined based on the major label of all neighbors.

5. Experiments

In this study, we introduced a novel extension of weighted k nearest neighbor algorithm. We conduct an experiment on a physiological status prediction case to evaluate the effectiveness of our proposed method. We select dataset of Physiological Data Modeling Contest (PDMC) described in next subsection.

5.1 Dataset description

PDMC dataset is provided by Body Media™ Inc. and released as contest subject dataset through physiological data modeling contest at ICML04 conference. Body Media ™ provided a rich dataset comprising several months of data from more than two dozen subjects [2]. Records of dataset have sensed data of 5 sensors that measure raw data of body features including skin temperature, heat flux, galvanic skin response, near-body temperature and subject's motion [3]. The subject's motion sensor generated 5 channels of raw data while other sensors generate only one channel. In addition to sensors' data two ordinal characteristics for each subject are added. In sum, the raw data contains 11 features. Data records are organized in more than 1400 time sessions and subjects determine their status during each minute of time sessions [3]. All data records labeled with 58 physiological statuses. Physiological labels are classified in different activity classes like walking, sleeping, watching TV, etc [2].

5.2 Procedure of Experiments

To evaluate our proposed algorithm we randomly select 300 records of each human subject (from 18 subject) to generate training set, contains 5400 records. Our sampled test set, as a subset of general test set of PDMC-04, contains 1800 records. All records in training and test sets are searched for abnormal cases e.g. negative temperature and cleaned. During PDMC-04 workshop, organizers asked participants to identify when a person is participating in context-1 class of activities. All labels of context-1 in training set are replaced with one and remain labels from other classes are zeroed out. We normalize and uniformly quantize training set records in 11 levels ([0, 10]) to generate discretized version of training set.

SRDA method is utilized to extract eigenvalues of discretized training set. Columns corresponded to top nonzero eigenvalues are selected to form reduced discretized training set. Figure 1. shows sorted values of elements of eigenvector of our training set. Based on these values we choose five columns associated to nonzero values.

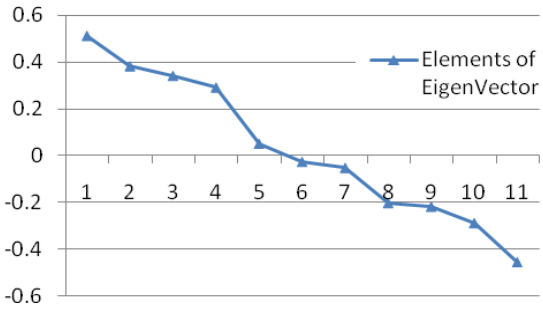

Figure 1. Sorted values of elements of eigenvector of our training set.

To this step, our discretized training set is reduced to a matrix with p=5 features from 11 discrete features. In next step, we categorize training set and extract different patterns out of 5400 records of reduced discretized training set. Category set matrix is filled with g=929 records of 5 features. All instances of 929 categories in whole training set are counted to generate total A-frequency values of categories. All records are partitioned into two classes (q=2). We counted all instances of categories that are appeared in each class to generate in-class A-frequencies. Then A-frequency matrix has 929 rows and 3 columns. Total/In-class A-probabilities are computed by normalizing total/in-class A-frequencies. For all categories 8 vectors of effectivity measures are calculated and form 8 vectors of size 929×1. Test set preprocessed in the same way as training set preprocessed. Test set normalized and quantized in 11 levels to discretize the test set. Features that are selected to appear in reduced training set are selected. Closeness of a sample from test set and a sample from

training set is measured by distance metrics thus, the first step in kNN algorithms is distance computation. Distances for all samples from test set and training set are computed based on the formulation of distance metrics. We select three metrics and for each distance a matrix of 1800×5400 elements are computed. To weight distances, we search all 929 categories to find an instance that is same as the sample of training set that is selected for measuring the closeness. Each effectivity measure of found category is multiplied by each of distances of the training set sample that is belonged to this category to weight them. Three distance metrics and 8 effectivity measures form 24 combinations of weighted distances. For all these combinations 24 weighted distance matrices are computed. Size of all weighted distance matrices are same as pure distance matrix, i.e. 1800×5400. As we commented before, we implemented kNN algorithm with k=3 (called 3NN). For each row of 24 weighted distance matrices we select three samples with minimum weighted distances. List of nearest neighbors is filled for every implementation of kNN with size of 1800×3. Major label for each rows of nearest neighbors is selected and assigned to that of sample.

5.3 Experimental results and discussion

To evaluate our method, we implement three versions of proposed algorithm based on three distance metrics. The performance of the proposed weighted 3NN was evaluated using accuracy criteria. Baseline accuracy of k-nearest neighbor without effectivity measures are presented in Table 2. Table 3 shows the accuracy rates of all compositions of distance metrics and effectivity measures.

Figure 2. compares weighted 3NN and baseline 3NN based on Euclidean distance metric and different effectivity measures. Figure 3. and Figure 4. illustrate baseline accuracy in comparison with accuracy of our method for complexity-invariant and edit distance metrics, respectively. Afterwards, we compare our three implementations of weighted 3NN and show the results in Figure 5. Figure 2. and Figure 3. show that weighted 3NN implemented using Euclidean and complexity-invariant distance metrics beats baseline 3NN algorithm under all effectivity measures. Weighted 3NN with Euclidean/complexity-invariant distance metrics performs the best under Shannon and Atkinson effectivity measures. The worst results occur when weighted 3NN applied Lorenz and Kullback effectivity measures. Weighted 3NN with edit distance carries out better than baseline 3NN under most of effectivity measures. Figure 5. show that weighted 3NN with Euclidean distance performs a little better than weighted 3NN with complexity-invariant distance measure. Euclidean distance as parameterized distance metric follows complexity invariant as a parameter free distance metric for time series. Figure 4 implies that for small streams with small set of discrete symbols (e.g. feature vectors) edit distance does not show meaningful difference between examples, thus weighting approach cannot adapt distances well.

Table 2. Accuracy rates of 3NN based on different distance metrics.

Euclidean Distance	Complexity-Invariant Distance	Edit Distance.
0.6672	0.6828	0.6683

Table 3. Accuracy rates for each effectivity measure and different distance metrics.

Effectivity Measure	Euclidean Distance	Complexity-Invariant Distance	Edit Distance
Total A-Probability	0.7239	0.7267	0.665
Shannon	0.7489	0.7344	0.6467
Gini	0.71	0.7206	0.7111
Lorenz	0.6772	0.6911	0.6967
Schutz	0.7089	0.7089	0.7072
Atkinson	0.7311	0.7383	0.7067
Whittaker	0.6961	0.7	0.695
Kullback	0.6917	0.6922	0.695

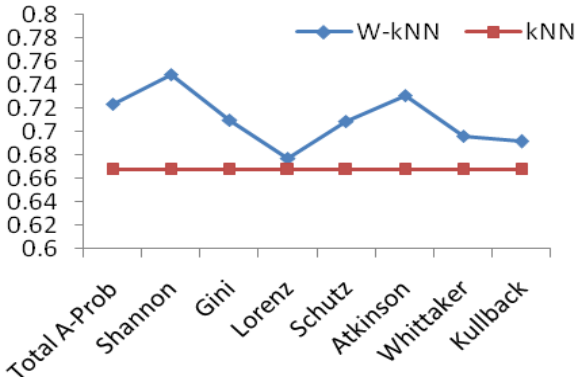

Figure 2. Comparison of weighted kNN and baseline kNN with Euclidean distance metric under effectivity measures.

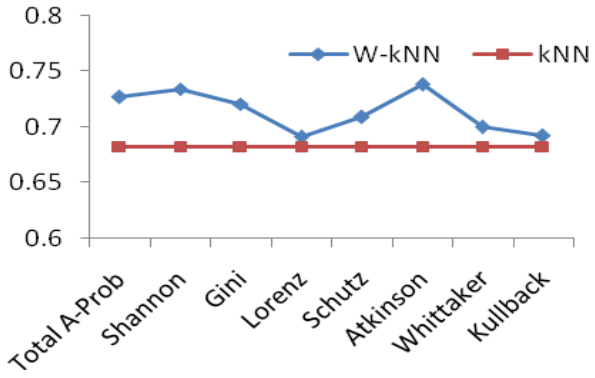

Figure 3. Comparison of weighted kNN and baseline kNN with Complexity-invariant distance metric under effectivity measures.

6. Conclusion

In this paper, we proposed a novel weighting approach for general k nearest neighbor based on the effectivity of classes in problem's space. Data categorization in training set were done based on the fact that quantization of large amount of data records in small set of bins leads to appearance of repeated patterns. Boundary points are more likely to belong to larger classes than smaller classes. Due to inverse relation between membership-probabilities of data points and their distances from center of classes, effectivity measures should decrease distances for neighbors of probable classes.

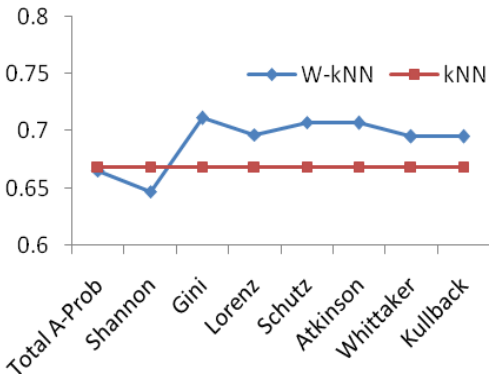

Figure 4. Comparison of weighted kNN and baseline kNN with Edit distance metric under effectivity measures.

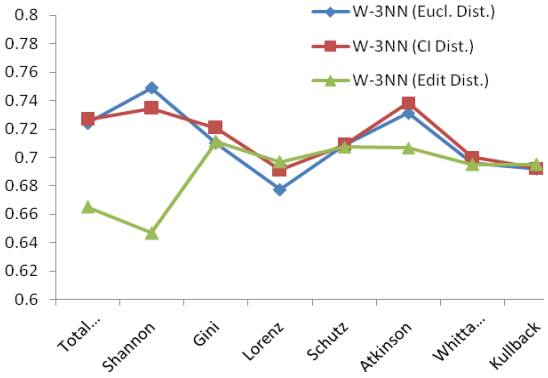

Figure 5. The Accuracy rates of weighted kNN implementations.

In our weighting approach records of training set are categorized and their probabilities of appearance in classes were computed. In-class probabilities were used in computing effectivity measures of categories. Effectivity measures of categories are their interestingness measures that reflect diversity of the category in different classes. In addition to our contribution on weighting calculation we applies weights before neighbor selection to move up some of neighbors might not be selected without weighting. Weighted kNN and baseline were implemented with three distance metrics. Results show that our algorithm increase accuracy of kNN and outline the baseline kNN by different compositions of distance metrics and effectivity measures. Accuracy rates of weighted kNN with Euclidean and complexity-invariant distance metrics are close together. Edit distance based weighted kNN performs better than baseline kNN with same distance metrics. Lower accuracy rate of Edit distance based weighted kNN in comparison with other implementation relies on low accuracy rate of edit distance in distinction of small strings with small set of symbols. We plan to evaluate our proposed algorithm on a set of benchmark datasets and multiple-label datasets, in near future.

7. REFERENCES

[1] Allen, M.G. 2005. Implantable micromachined wireless pressure sensors: approach and clinical demonstration. *The Second International Workshop on Wearable and Implantable Body Sensor Networks* (2005), 40-43.

[2] Andre, D. 2004. ICML competition and workshop on Physiological Data Modeling. (2004).

[3] Andrews, S., Cai, L., Gondek, D., Greenwald, A., Grollman, D., Jonsson, A.M., Hall, K., Lease, M., Bryant, N. g., Raiti, J., Sweetser, V. and Turner, J. 2004. Astrology: The Study of Astro Teller. *International Conference on Machine Learning Physiological Modeling Contest* (2004).

[4] Batista, G.E.A.P.A., Wang, X. and Keogh, E.J. 2011. A Complexity-Invariant Distance Measure for Time Series. *The 11th SIAM International Conference on Data Mining (SDM 2011)* (2011), 699-710.

[5] Bock, J. and Gough, D.A. 1998. Toward Prediction of Physiological State Signals in Sleep Apnea. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*. 45, 11 (1998), 1332-1341.

[6] Celi, L.A., Hinske, L.C., Alterovitz, G. and Szolovits, P. 2008. An artificial intelligence tool to predict fluid requirement in the intensive care unit: a proof-of-concept study. *Critical Care*. 12, 6 (2008), R151.

[7] Chang, Y.H. 2005. Emotion Recognition and Evaluation of Mandarin Speech Using Weighted D-KNN Classification. *Computational Linguistics and Speech Processing XVII (ROCLING XVII)* (2005).

[8] Chen, X., Xu, D., Zhang, G. and Mukkamala, R. 2009. Forecasting acute hypotensive episodes in intensive care patients based on peripheral arterial blook pressure waveform. *Computers in Cardiology (CinC)* (2009).

[9] Deng, C., Xiaofei, H. and Jiawei, H. 2008. SRDA: An Efficient Algorithm for Large Scale Discriminant Analysis. *IEEE Transactions on Knowledge and Data Engineering*. 20, 1 (2008), 1-12.

[10] Dudani, S.A. 1976. The Distance-Weighted K-Nearest-Neighbor Rule. *IEEE Transaction on System, Man and Cybernetics*. (1976), 325-327.

[11] Geng, L. and Hamilton, H.J. 2006. Interestingness Measures for Data Mining: A Survey. *ACM Computing Surveys*. 38, 3 (2006).

[12] Hunink, M.G.M. and Glasziou, P.P. 2006. *Decision Making in Helath and Medicine - Integrating Evidence and Values*. Cambridge University Press.

[13] Imhoff, M., Kuhls, S., Gather, U. and Fried, R. 2009. Smart alarms from medical devices in the OR and ICU. *Best Practice & Research Clinical Anaesthesiology*. 23, 1 (2009), 39-50.

[14] Jeffreys, H. 1948. *Theory of Probability*. Clarendon Press.

[15] Khan, P., Hussain, A. and Kwak, K.S. 2009. Medical Applications of Wireless Body Area Networks. *International*

Journal of Digital Content Technology and its Applications. 3, 3 (2009), 185-193.

[16] Lorincz, K., Kuris, B., Ayer, S.M., Patel, S., Bonato, P. and Welsh, M. 2007. Wearable Wireless Sensor Network to Assess Clinical Status in Patients with Neurological Disorders. *Information Processing in Sensor Networks (IPSN)* (2007).

[17] Luo, G. 2011. Triggers and Monitoring in Intelligent Personal Health Record. *Journal of Medical Systems*. (2011), 1-17.

[18] Osheroff, J.A., Teich, J.M., Middleton, B., Steen, E.B., Wright, A. and Detmer, D.E. 2007. A roadmap for national action on clinical decision support. *J. Med. Inform. Assoc*. 14, 2 (2007), 141–145.

[19] Rokach, L. and Maimon, O. 2008. *DATA MINING WITH DECISION TREES:Theory and Applications*. World Scientific Publishing.

[20] Tran, T.N., Wehrens, R. and Buydens, L.M. 2006. KNN-kernel density-based clustering for high-dimensional multivariate data. *Computational Statistics Data Analysis*. 51, 2 (2006), 513-525.

[21] Varshney, U. 2007. Pervasive healthcare and wireless health monitoring. *Mob. Netw. Appl.* 12, (2007), 113-127.

[22] Vassanyi, I., Kozmann, G., Banhalmi, A., Vegso, B., Kosa, I., Dulai, T., Tarjanyi, Z., Tuboly, G., Cserti, P. and Pinter, B. 2011. Applications of medical intelligence in remote monitoring. 169, (2011), 671-675.

[23] Vivencio, D.P., Hruschka, E.R.J., Nicoletti, M.C., dos Santos, E.B. and Galvio, S.D.C.O. 2007. Feature-weighted k-Nearest Neighbor Classifier. *IEEE Symposium on Foundations of Computational Intelligence (FOCI 2007)* (2007).

[24] Yu, K. and Ji, L. 2002. Karyotyping of Comparative Genomic Hybridization Human Metaphases Using Kernel Nearest-Neighbor Algorithm. *Cytometry*. 48, (2002), 202-208.