Nearest Neighbor Classification with Locally Weighted Distance for Imbalanced Data

Zahra Hajizadeh, Mohammad Taheri, and Mansoor Zolghadri Jahromi

***Abstract*—The datasets used in many real applications are highly imbalanced which makes classification problem hard. Classifying the minor class instances is difficult due to bias of the classifier output to the major classes. Nearest neighbor is one of the most popular and simplest classifiers with good performance on many datasets. However, correctly classifying the minor class is commonly sacrificed to achieve a better performance on others. This paper is aimed to improve the performance of nearest neighbor in imbalanced domains, without disrupting the real data distribution. Prototype-weighting is proposed, here, to locally adapting the distances to increase the chance of prototypes from minor class to be the nearest neighbor of a query instance. The objective function is, here, G-mean and optimization process is performed using gradient ascent method. Comparing the experimental results, our proposed method significantly outperformed similar works on 24 standard data sets.**

***Index Terms*—Gradient ascent, imbalanced data, nearest neighbor, weighted distance.**

I. Introduction

In recent years, the classification problem in imbalanced data sets has been identified as an important problem in data mining and machine learning, because the imbalanced distribution is pervasive in most of real-world problems. In these datasets, the number of instances of one of the classes is much lower than the instances of the others [1]. The imbalance ratio may be on the order of 100 to one, 1000 to one, or even higher [2].

Imbalanced data set appears in most of the real world domains, such as text classification, image classification, fraud detection, anomaly detection, medical diagnosis, web site clustering and risk management [3]. We worked on the binary class imbalanced data sets, where there is only one positive and one negative class. The positive and negative classes are respectively considered as the minor and major classes.

If the classes are nested with high overlapping, separating the instances of different classes is hard. In these situations, the instances of minor class are neglected in order to correctly classify the major class, and to increase the classification rate. Hence, learning algorithms, which train the parameters of the classifier to maximize the classification rate, are not suitable for the case of imbalanced datasets.

Normally in the real applications, detecting the instances from the minor class is more valuable than others [4]. A good performance on minor instances may not be achieved even with having the maximum classification rate. This is why; some other criteria have been proposed to measure the performance of a classifier on imbalanced datasets. These criteria measure the performance of the classifier on both of minor and major classes. In this paper, G-mean is used and described in section 3. Here, this criterion is also used as the objective function instead of the pure classification rate.

There are several methods to tackle the problems of imbalanced data sets. These methods are grouped into two categories: internal and external approaches. In the former approach, a new algorithm is proposed from scratch, or some existed methods are modified [5], [6]. In external approaches, data is preprocessed in order to reduce the impact of the class imbalance [7], [8]. Internal approaches are strongly dependent on the type of the algorithm, while external approaches (sampling methods) modify the data distribution regardless of the final classifier. The major drawbacks of the sampling method are loss of useful data, over-fitting and over generalization. In this paper an internal approach is proposed based on modifying the learning algorithm in an adaptive distance nearest neighbor classifier.

Nearest neighbor classifier (NN) has been identified as one of the top ten most influential data mining algorithms [9] due to its simplicity and high performance. The classification error rate of nearest neighbor is not more than twice the Bayes [10] where the number of training instances is sufficiently large. Even in nearest neighbor classifier that has no training phase, without any priority knowledge of the query instance, it is more likely that the nearest neighbor is a prototype from the major class. This is why, this classifier has not a good performance to classify the instances of the minor class, especially where the minor instances are distributed between the major ones [11].

In this paper, we proposed an approach to improve the nearest neighbor algorithm on imbalanced data. This approach has a good performance on the classified minor class instances, and the major class instances are acceptably classified as well. According to data distribution, in the proposed method, a weight is assigned to each prototype. Distance of each query instance from a prototype is directly related to the weight of the prototypes. With this approach, the prototypes with smaller weights have more chances to be the nearest neighbor of the new query instance. This weighting is done in such a way that increases the performance of nearest neighbor based on G-mean.

In order to analyze the experimental results, 24 standard benchmark datasets from UCI repository of machine learning databases [12] are used. For multi-class data sets, the class

Manuscript received October 30, 2013; revised January 25, 2014.

The authors are with the Department of Computer Science and Engineering and Information Technology, School of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran (email: z-hajizadeh, taheri@cse.shirazu.ac.ir, zjahromi@shirazu.ac.ir).

DOI: 10.7763/IJCCE.2014.V3.296

with the smallest size is considered as the positive class, and the rest of instances are labeled as the negative class. In comparison of nearest neighbor with some other well-known algorithms such as SVM and MLP, nearest neighbor can be easily used to support the multi-class datasets, which is out of the scope of this paper.

The rest of this paper is organized as follows: In Section II, the related works done in the past are described. In Section III, the evaluation measurements in imbalanced domains are briefly described. In Section IV, the proposed approach is presented. The experiments are reported in Section V and the paper is concluded in Section VI.

II. Related Works

Various solutions have been proposed to solve the problems of imbalanced data. These solutions include a wide variety of two different approaches, modified algorithms and preprocesses. Methods that preprocess on the data are known as sampling techniques to oversample instances in the minor class (sample generation) or to under-sample in the major one (sample selection) [13]. One of the earliest and classic works, called SMOTE method [14], increases the number of minor class instances by creating synthetic samples. This method is based on the nearest neighbor algorithm. The minor class is over sampled by generating new samples along the line segments connecting each instance of the minor class to its k-nearest neighbors. In order to improve the SMOTE method, the safe-level-SMOTE [15] method has been introduced, in which the samples have different weights in generating synthetic samples. The other types of algorithms have focus on extending or modifying the existing classification algorithms such that they can be more effective in dealing with imbalanced data. HDDT [16] and CCPDT [17] are examples of these methods, which are modified versions of decision tree classifiers. Over the past few decades, kNN algorithm is widely studied, and has been used in many fields. The kNN classifier classifies each unlabeled sample by the major label of its k nearest neighbors in the training dataset. Weighted Distance Nearest Neighbor (WDNN) [18] is a recent work on prototype reduction based on retaining the informative instances and learning their weights to improve the classification rate on training data. The WDNN algorithm is well formulated, and shows encouraging performance; however, it can only work with k=1 in practice. WDkNN [19] is another recent approach which attempts to reduce the time complexity of WDNN, and extends it to work for values of k greater than 1.

Chawla and Liu, [20] in one of their recent works, presented a new approach named Class Confidence Weighted (CCW) to solve the problems of imbalanced data. While conventional kNN only uses prior information for classification of samples, CCW converts prior to posterior, thus operates as likelihood in Bayes theory, and increases its performance. In this method, weights are assigned to samples by the Mixture Model and Bayesian Networks method.

Based on the idea of informative-ness, two different versions of kNN are proposed by Yang Song and others [21]. According to them, a sample is treated to be informative, if it is close to the query instance, and far away from the samples with different class labels. LI-KNN is one of the proposed versions, which takes two parameters k and I. It first finds the k nearest neighbor of the query instance, and then among them it finds the I most informative samples. Class label is assigned to the query instance based on its informative samples. They also demonstrated that the value of k and I have very less effect on the final result. GI-KNN is the other version which works on the assumption that some samples are more informative then others. It first finds global informative samples, and then assigns a weight to each of the samples in training data based on their informative-ness. It then uses weighted Euclidean metric to calculate distances.

Paredes and Vidal [22] have proposed a sample reduction and weighted method to improve nearest neighbor classifier, called LPD (Learning Prototype and Distance), which has received many attentions. The algorithm simultaneously trains both a reduced set of prototypes and a suitable weight for associated prototypes. Such that the defined objective function, which is the error rate, is optimized on the training samples by using gradient based decreasing method. In this way, the weight is individually assigned to each feature. In the test stage, the reduced set with related weights is used.

III. Evaluation in Imbalanced Domains

The measures of the quality of classification are built from a confusion matrix (shown in Table I) which records correct and incorrect recognized samples for each class.

TABLE I: Confusion Matrix for a Two-Class Problem

	Positive prediction	**Negative prediction**
Positive class	True positive (TP)	False negative (FN)
Negative class	False positive (FP)	True negative (TN)

Accuracy defined in (1) is the most used empirical measure, which does not distinguish between the number of correct labels of different classes. This issue may lead to erroneous conclusions in imbalanced problems. For instance, a classifier may not correctly cover a minor class instances even it obtains an accuracy of 90% in a data set.

$$ACC = \frac{TP + TN}{TP + FN + FP + TN}. \quad (1)$$

For this reason, more correct metrics are considered in addition to using accuracy. Sensitivity (2) and specificity (3) are two common measures which approximate the probability of the positive/negative label being true. In other words, they assess the effectiveness of the algorithm on a single class.

$$sensitivity = \frac{TP}{TP + FN}. \quad (2)$$

$$specificity = \frac{TN}{FP + TN}. \quad (3)$$

In this paper, the geometric mean of the true rates is used as the metric [7]. It can be defined as:

$$G - mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{FP + TN}}. \quad (4)$$

This metric attempts to maximize the accuracy of one of the two classes with a good balance.

IV. Method

Let $T=\{x_i,...,x_N\}$ be a training set, consists of N training instances. Therefore, each training instance will be denoted either "$x \in T$" or "x_i, $1 \le i \le N$". The index of training instance x in T is denoted as index(x), defined as index(x)=i iff $x=x_i$. Let C be the number of classes.

The weighted distance from an arbitrary training instance or test instance x to other training instance $x_i \in T$ is defined as:

$$d_w(x, x_i) = w_i \parallel x - x_i \parallel_2 . \quad (5)$$

where w_i is an associated weight to samples. The W vector can be represented as $W=\{w_i, 1 \le i \le N\}$. As will be discussed later, other distance could also be used because it does not affect how to determine the weights. As shown in (5), when the weighting factor for an instance is greater than the other instances, its distance gets farther to the others. Because of this, the probability of being chosen as the instance reduces as the nearest neighbor goes farther.

To find the suitable weight vector, training samples are operated in a way which the objective function is maximized. In this paper, the objective function is G-mean measured as in (4). We could easily use ln(G-mean) as the objective function without losing the optimal reply. Therefore, the objective function is defined as:

$$J(W) = \sum_{k=1}^{c} \ln[ACC_k]. \quad (6)$$

The *ACC* is defined as:

$$ACC = \frac{1}{n_c} \sum_{x \in c} step\left(\frac{d_w(x, x_i^{\neq})}{d_w(x, x_i^{=})} \right). \quad (7)$$

And the *step* function is defined as:

$$step(z) = \begin{cases} 0 & if \quad z \le 1 \\ 1 & if \quad z > 1 \end{cases}.$$

$x_i^{=} \in T$ and $x_i^{\neq} \in T$, are respectively the *same-class* and *different-class* nearest neighbors of the desired training data. The incremental gradient method is used in order to maximize the objective function. This requires J to be differentiable with respect to all the parameters to be optimized; i.e., w_i, $1 \le i \le N$. Thus, the step function will be approximated by using a *sigmoid* function, defined as:

$$\varphi_\beta(z) = \frac{1}{1 + e^{\beta(1-z)}}. \quad (8)$$

With this approximation, the proposed *ACC* becomes:

$$ACC \approx \frac{1}{n_c} \sum_{x \in c} \varphi_\beta(r(x)). \quad (9)$$

where

$$r(x) = \frac{d_w(x, x_i^{\neq})}{d_w(x, x_i^{=})}. \quad (10)$$

The derivate of $\varphi_\beta(.)$ will be used:

$$\varphi'_\beta(z) = \frac{d_{\varphi_\beta(z)}}{dz} = \beta(1 - \varphi_\beta(z))\varphi_\beta(z). \quad (11)$$

where $\varphi'_\beta(z)$ represents the Dirac delta function if β is large, whereas it is approximately constant for a wide range of values of z if it is small. Partial derivative of (6) with respect to W is as follows:

$$\frac{\partial J}{\partial w_i} \approx \frac{\beta}{w_i} \sum_{x_j \in T} \frac{(1 - \phi_\beta(r(x_j)))\phi_\beta(r(x_j))r(x_j)}{L(x_j)} S_i(x_j). \quad (12)$$

where

$$S_i(x_j) = \begin{cases} 1 & x_i = x_j^{\neq} \\ -1 & x_i = x_j^{=} \\ 0 & otherwise \end{cases}$$

And

$$\forall c = 1, 2 \quad \forall x_j \in T \quad L(x_j) = \sum_{x \in c} \phi_\beta(r(x))$$

By using (12) the weights are updated in several iterations until the proper weight is achieved. By visiting each training instance we could calculate gradient for all training instances with the *same-class* and *different-class* nearest neighbors of it. Then we updated weighs according to the following formula:

$$\forall x_i \in T \quad w_i^{new} = w_i^{old} + \alpha \frac{\partial J}{\partial w_i}.$$

Which w_i^{new} and w_i^{old}, the new and former weights are respectively associated with the ith instance. α is the learning step factor, that sets by line-search technique [23]. Proposed algorithm is shown in Fig. 1.

The main advantage of this algorithm is its ability to correct the substantial bias to major class in existing kNN algorithms, and correctly classify the minor class instances, which are very important.

```
Algorithm (T, W, β, α, ε) {
  //T: training set;  W: initial weights;
  //β: sigmoid slope;  α: learning factor;
  // ε: small constant.
  λ' = ∞;   λ = J(W);   W' = W;
  while(|λ' - λ| > ε) {
    λ' = λ;
    for all x ∈ T {
      x⁼ = FINDNN_SAMECLASS(W, x);
      x≠ = FINDNN_DIFFCLASS(W, x);
      i = index(x⁼);
      k = index(x≠);
      T(x) = (1-φβ(r(x))).φβ(r(x)).r(x)/L(x);
      w'i = w'i – α.β/wi.T(x);
      w'k = w'k + α.β/wk.T(x);
    }
    W = W';   λ = J(W);
    If(λ<λ'){
       α=α/2;
    }
}
  return(W);
}
```

Fig. 1. Summary of the proposed method.

V. Experimental Results

Proposed algorithm tested on 24 imbalanced data sets from UCI repository of machine learning databases. We used the IR [24], defined as the ratio of the number of instances of the major class and the minor class, to demonstrate distinction among imbalanced data sets. The data sets are highly imbalanced where there are no more than 10% of positive instances in the whole data set compared to the negative ones, or in other words when IR is higher than 9. Here, we used imbalanced data sets with IR higher than 9. Summary descriptions for imbalanced data sets are shown in Table II, which are in descending order based on IR. For each data set, the number of examples (#Ex.), number of attributes (#Atts.) and percentage of major and minor classes (%Class (min.,maj.)) are demonstrated. Because the UCI data sets are small, K-fold cross validation (KCV) is used to obtain the classification results. Each data set is divided into K blocks using K−1 blocks as a training set, T, and the remaining block as a test set. Here, the number of blocks is set to K=10; it repeated 10 times, therefore, the final results are the average of the 100 different results.

TABLE II: Summary Description for Imbalanced Data Sets

Data-set	#Ex	#Atts	%Class (min., maj.)	IR
Glass5	214	9	(4.20,95.79)	22.77
shuttle2vs4	129	9	(4.65,95.35)	20.5
abalone918	731	8	(5.74,94.25)	16.40
ecoli4	336	7	(5.95,94.05)	15.80
Glass4	214	9	(6.07,93.92)	15.46
ecoli034vs5	300	7	(6.66,93.33)	14
ecoli0146vs5	280	7	(7.14,92.86)	13
ecoli0147vs56	332	7	(7.53,92.47)	12.28
Glass2	214	9	(7.94,92.05)	11.59
glass0146vs2	205	9	(8.29,91.71)	11.05
ecoli01vs5	240	7	(8.33,91.66)	11
glass06vs5	108	9	(8.33,91.66)	11
ecoli0147vs2356	336	7	(8.63,91.37)	10.58
ecoli067vs5	220	7	(9.09,90.90)	10
Vowel0	988	13	(9.11,90.89)	9.98
ecoli0347vs56	257	7	(9.73,90.27)	9.28
ecoli0346vs5	205	7	(9.76,90,24)	9.25
glass04vs5	92	9	(9.78,90.22)	9.22
ecoli0267vs35	224	7	(9.82,90.18)	9.18
ecoli01vs235	244	7	(9.83,90.16)	9.16
ecoli046vs5	203	7	(9.85,90.15)	9.15
glass015vs2	172	9	(9.88,90.12)	9.11
ecoli067vs35	222	7	(9.91,90.09)	9.09
yeast2vs4	514	8	(9.92,90.08)	9.08

The sigmoid derivative is almost constant for small values of β, and the algorithm approximately learns the same regardless of the different values of $r(x)$. For large values of β, when the distance ratio or $r(x)$ is very close to 1, either the learning happens, or it never does. A suitable β value should let the classifier learn from misclassified samples, but should prevent learning from outliers, which the $r(x)$ value is very big. In according to do tests, β should be equal to 10 and set all initial w_i=1. The learning factor (α) is set by line-search technique [23].

Our algorithm was compared with 6 other algorithms. In this paper, the SMOTE implementation suggested k=5. We used the Euclidian distance in order to produced synthetic samples, and balanced both classes to the 50% distribution.

SMOTE and random under-sampling methods use nearest neighbor classifier to classify query instances. LPD [22] has been recognized as a very successful method. In this paper, primary samples are 5% of training instances randomly chosen while equal samples are chosen from each class to be ignored the effect of imbalanced data. Because of the LPD's aim is to increase classification accuracy, it does not perform well on imbalanced data. The results are shown in Table III. NN is the nearest neighbor algorithm without considering weight for samples, and NWKNN [25] represents a weighting method to improve the kNN algorithm on imbalanced data.

The Friedman was used to find significant differences among the results obtained by the studied methods [26]. The Friedman test is a nonparametric statistical method for testing whether all the algorithms are equivalent over various data sets.

If there are significant differences between the algorithms, we used post-hoc test [27] to find out which algorithms actually differed. In this paper, an improved Friedman test proposed by Iman and Davenport [28] is used; therefore Bonferroni-Dunn test [29] is used as the post-hoc test method.

The Friedman test compares an average of classifiers ranks. The smaller rank indicates higher grade and better performance. Table III shows Friedman test results. The ranks of classifiers on each data set are shown in parentheses in Table III. Our algorithm is the Base classifier; therefore; the average ranks differences between the Base classifier and the other ones are calculated. CD indicates critical difference [27], which is determined by algorithm numbers, data sets, and critical value $q\alpha$ that $q\alpha$=2.638. If the average ranks differences of two methods are larger than CD, there is a significant difference between them. A checkmark sign indicates significant difference between the methods. According to the average ranks, our algorithm has the highest grade. Friedman test rejected the null hypothesis, which indicates significant difference between the methods. Our algorithm significantly has better performance than the other ones.

VI. Conclusion

A new learning technique based on gradient ascent is proposed to improve the nearest neighbor algorithm on imbalanced data. A number of experiments involving a large collection of standard benchmark imbalanced data sets showed the good performance of this technique.

The importance of the slope of the sigmoid function has been studied both conceptually and empirically. The slope of the sigmoid function has an important role to control the learning process. Adequate values (typically around 10) help reducing the impact of outliers.

Future works could focus on local optimization, making dynamic β value during training, investigating other weighting methods and using the other objective functions.

TABLE III: COMPARISON OF THE G-MEAN RESULTS FOR 7 ALGORITHMS FOR TEST

Data-set	SMOTE	Random under-sampling	NWKNN	NN	LPD	Our algorithm with prototype selection	Our algorithm without prototype selection
Glass5	68.76(5)	86.15(2)	49.75(7)	71.26(4)	53.94(6)	77.36(3)	**88.04(1)**
shuttle2vs4	99.57(2)	82.38(6)	60.00(7)	90.00(5)	96.44(3)	90.76(4)	**100(1)**
abalone918	62.55(3)	60.00(4)	05.00(7)	28.05(5)	14.92(6)	**64.08(1)**	62.57(2)
ecoli4	90.14(4)	91.96(2)	71.88(7)	84.84(6)	87.47(5)	**92.20(1)**	90.42(3)
Glass4	88.47(2)	80.30(4)	43.65(7)	77.87(5)	73.57(6)	81.74(3)	**88.71(1)**
ecoli034vs5	**97.07(1.5)**	87.69(5)	90.53(4)	83.05(7)	91.81(3)	86.94(6)	**97.07(1.5)**
ecoli0146vs5	87.95(2)	87.08(4)	82.04(7)	83.90(6)	86.01(5)	**88.29(1)**	87.37(3)
ecoli0147vs56	85.80(4)	87.39(3)	75.14(7)	83.29(6)	85.19(5)	87.60(2)	**88.13(1)**
Glass2	39.05(4)	48.30(3)	00.00(7)	33.40(5)	06.69(6)	60.17(2)	**64.06(1)**
glass0146vs2	41.98(4)	49.28(3)	00.00(7)	35.54(5)	06.37(6)	**66.68(1)**	56.04(2)
ecoli01vs5	84.67(7)	89.59(2.5)	87.59(4)	84.77(6)	85.65(5)	89.59(2.5)	**90.36(1)**
glass06vs5	87.95(3)	76.38(5)	49.49(7)	**89.05(1)**	59.10(6)	76.63(4)	88.94(2)
ecoli0147vs2356	80.27(5)	**84.31(1)**	59.79(7)	79.58(6)	82.29(3)	80.85(4)	83.15(2)
ecoli067vs5	84.23(3)	82.69(4)	62.25(7)	79.02(6)	82.37(5)	85.40(2)	**86.65(1)**
Vowel0	**100(2)**	91.21(7)	97.66(5)	**100(2)**	97.95(4)	97.60(6)	**100(2)**
ecoli0347vs56	83.97(4)	83.50(5)	71.56(7)	83.37(6)	85.26(3)	**87.01(1)**	86.90(2)
ecoli0346vs5	87.81(2)	85.43(4)	77.74(7)	83.29(6)	85.39(5)	87.01(3)	**90.02(1)**
glass04vs5	90.00(2)	79.73(5)	68.01(7)	70.57(6)	84.98(3)	80.85(4)	**99.35(1)**
ecoli0267vs35	76.08(5)	76.02(6)	65.86(7)	81.57(2)	80.75(3)	79.77(4)	**85.20(1)**
ecoli01vs235	82.76(5)	83.13(4)	73.12(7)	85.62(2)	77.94(6)	83.55(3)	**86.77(1)**
ecoli046vs5	88.00(2)	86.62(4)	76.95(7)	85.37(5)	84.52(6)	**88.53(1)**	87.51(3)
glass015vs2	42.96(4)	43.70(3)	07.07(6)	33.77(5)	05.25(7)	52.08(2)	**53.15(1)**
ecoli067vs35	68.09(6)	78.80(2)	45.42(7)	78.54(3)	75.04(4)	72.65(5)	**79.43(1)**
yeast2vs4	85.38(3)	85.06(4)	66.16(7)	81.63(5)	76.67(6)	86.87(2)	**90.87(1)**
Average Rank	3.52	3.83	6.62	4.75	4.87	2.81	1.52
Friedman Test	*Reject*						
	2	**2.31**	**5.1**	**3.23**	**3.35**	1.29	**Base**
	✓	✓	✓	✓	✓		
***CD=1.64**							

REFERENCES

[1] A. Fernández, M. J. del Jesus, and F. Herrera, "Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets," *Int. J. Approx. Reasoning*, vol. 50, no. 3, pp. 561-577, Mar. 2009.

[2] E. Kriminger and C. Lakshminarayan, "Nearest neighbor distributions for imbalanced classification," in *Proc. IEEE World Congress on Computational Intelligence*, June 10-15, 2012, pp. 1-5.

[3] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263-1284, 2009.

[4] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas. "Handling imbalanced datasets: a review," *GESTS International Transaction on Computer Science and Engineering*, vol. 30, 2006.

[5] R. Barandela, J. S. Sánchez, V. García, and E. Rangel, "Strategies for learning in class imbalance problems," *Pattern Recognition*, vol. 36, no. 3, pp. 849-851, 2003.

[6] L. Xu, M. Y. Chow, and L. S. Taylor, "Power distribution fault cause identification with imbalanced data using the data mining-based fuzzy classification ealgorithm," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 164-171, 2007.

[7] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behaviour of several methods for balancing machine learning training data," *SIGKDD Explorations*, vol. 6, no. 1, pp. 20-29, 2004.

[8] A. Estabrooks, T. Jo, and N. Japkowicz, "A multiple resampling method for learning from imbalanced data-sets," *Computational Intelligence*, vol. 20, no. 1, pp. 18-36, 2004.

[9] X. Wu *et al*., "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no.1, pp. 1-37, 2008.

[10] G. Bhattacharya, K. Ghosh, and A. S. Chowdhury, "An affinity-based new local distance function and similarity measure for kNN algorithm," *Pattern Recognition Letters*, vol. 33, pp. 356-363, 2012.

[11] N. Japkowicz, in *Proc. the AAAI'2000 Workshop on Learning from Imbalanced Data Sets*, 2000, AAAI Tech Report.

[12] A. Asuncion and D. Newman. (2007). UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[13] S. Garcia and F. Herrera, "Evolutionary under sampling for classification with imbalanced datasets: proposals and taxonomy," *Evolutionary Computation*, vol. 17, no. 3, pp. 275-306, 2009.

[14] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, "Smote," *Journal of Artificial Intelligence Research*, vol. 16, no. 1, pp. 321-357, 2002.

[15] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap, "Safe-level-smote: safe level synthetic minor over-sampling technique for handling the class imbalanced problem," in *Proc. PAKDD'09*, vol. 5476, pp. 475-482.

[16] D. A. Cieslak and N. V. Chawla, "Learning decision trees for unbalanced data," in *Proc. ECML*, 2008, vol. 5211, pp. 241-256.

[17] D. Cieslak, W. Liu, S. Chawla, and N. Chawla, "A robust decision tree algorithms for imbalanced data sets," in *Proc. the Tenth SIAM International Conference on Data Mining*, 2010, pp. 766-777.

[18] M. Z. Jahromi, E. Parvinnia, and R. John, "A method of learning weighted similarity function to improve the performance of nearest neighbor," *Inf. Sci*., vol. 179, pp. 2964-2973, 2009.

[19] T. Yang, L. Cao, and C. Zhang, "A novel prototype reduction method for the K-nearest neighbor algorithm with $K \geq 1$," in *Proc. PAKDD*, 2010, vol. 6119, pp. 89-100.

[20] W. Liu and S. Chawla, "Class confidence weighted knn algorithms for imbalanced datasets," in *Proc. PAKDD*, 2011, vol. 6635, pp. 345-356.

[21] Y. Song, J. Huang, D. Zhou, H. Zha, and C. L. Giles, "Iknn: Informative k-nearest neighbor pattern classification," in *Proc. PKDD*, 2007, vol. 4702, pp. 248-264.

[22] R. Paredes and E. Vidal, "Learning prototypes and distances: A prototype reduction technique based on nearest neighbor error minimization," *Pattern Recognition*. vol. 39, no. 2, pp. 171–179, 2006.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge, U.K.: Cambridge Univ. Press, 2004.

[24] A. Orriols-Puig and E. Bernadó-Mansilla, "Evolutionary rule-based systems for imbalanced data-sets," *Soft Computing*, vol. 13 no. 3, pp. 213 – 225, 2009.
[25] S. Tan, "Neighbor-weighted K-nearest neighbor for unbalanced text corpus," *Expert System with Application*, vol. 28, pp. 667-671, 2005.
[26] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675-701, 1937.
[27] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *The Journal of Machine Learning Research*, vol. 7, pp. 1-30, 2006.
[28] R. L. Iman and J. M. Davenport, "Approximations of thecritical region of the friedman statistic," *Communications in statistics*, vol. 9, no. 6, pp. 571-595, 1980.
[29] O. J. Dunn, "Multiple comparisons among means, "*Journal of the American Statistical Association*, vol. 56 no. 293, pp. 52-64, 1961.

Zahra Hajizadeh was born in Shiraz, Iran. She received the B.Sc. degree in computer software engineering from the Azad University of Shiraz, Shiraz, Iran, in 2008. She is currently a M.Sc. student in artificial intelligence at Shiraz University. Her research interests include class imbalance learning, machine learning and data mining.

Mohammad Taheri was born in Sirjan, Iran, in 1983. He received the B.Sc. and M.Sc. degrees in computer science and engineering from Shiraz University, Shiraz, Iran, in 2005 and 2008 respectively. He is currently pursuing the Ph.D. degree in the same major and university.

From 2004, he worked as a researcher and manager in Pardazeshgaran company. He has many publications in the field of parameter learning in classifiers and also he has reviewed many papers about SVM and Fuzzy systems sent from well-known journals and transactions (e.g. IEEE Trans. on Fuzzy Systems).

Mansoor Zolghadri Jahromi received the B.Sc. degree in mechanical engineering from Shiraz University, Shiraz, Iran, in 1979, M.Sc. degree and Ph.D. degree both in instrumentation and control engineering from University of Bradford, West Yorkshire, England, in 1982 and 1988 respectively. He is currently a professor in the Department of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran. His research interests include control systems, Intelligent control systems, Instrumentation, information retrieval systems, computer networks and data communications. He has published many refereed papers and book chapters on these fields.