Bayesian-Based Instance Weighting For More Noise-Tolerant Instance-Based Learners

Khalil El Hindi, Bayan Abu Shawar

Abstract**—** Instance-Based learners are simple, yet, effective learners. They classify a new instance based on the k most similar instances which makes them sensitive to noise in training data sets. Obtaining good classification accuracy may, therefore, require cleaning the data sets using labor-extensive or computationally expensive data cleaning procedures. In this work, we present some Bayesian-based instance weighting techniques to make such learners more tolerant to noise. The basic idea is that typical or classical instances should be given more weight or voting power than less typical or noisy instances. We present three techniques to determine instance weights that are based on the conditional probability of an instance belonging to its actual class and not to another class. Our empirical results using the kNN algorithm shows that all presented techniques are effective in making the kNN more tolerant to noise. These results suggest that these techniques can be used with instance based learners instead of more expensive data cleaning procedures.

Keywords**—**Machine Learning, Instance-Based Learning, kNN algorithm, Instance Weighting, Naïve Bayesian.

I. INTRODUCTION

Instance-based learners (1), such as the K Nearest Neighbor algorithm kNN, are simple machine learning algorithms that proved to be very effective in many classification tasks such as text classification (2) and action recognition (3).

They classify a new unclassified instance based on its similarity to other classified instances. Similarity is measured using a distance function that calculates the distance between two instances. Once determined, each of the k most similar instance counts as one vote for its class, where k is typically an odd integer such as 3. The class that gets the majority of votes is taken to be the class of the new instance. Usually, instances have equal votes or their votes are assigned weights that are inversely proportional to their distances from the new instance. This gives closer instances more influence on the classification results than farther instances. However, this makes them sensitive to noisy instances which are very common in real life data sets. That is why the kNN uses an odd number for k, such as 3 or 5 rather than 1. The first nearest neighbor algorithm (kNN with k set to 1) is more sensitive to noise. Some noise filtering techniques can also be used to eliminate noisy instances from training data (see (4) for a good review) as a preprocessing step.

This work, presents three Bayesian-based vote weighting techniques that make the kNN more tolerant to noise in the training data. We assign a weight for each instance depending on the conditional probability that the instance truly belongs to its class and not to some other class. The intuition is the more certain we are that an instance belongs to its class and not to some other class, the more voting weight it is given. Noisy instance or untypical (exceptional) instances tend to get lower weights than more typical instances. Since the methods are based on global information rather than local information they are expected to be more tolerant to noise.

Our empirical results show that these techniques substantially improve the classification accuracy of the kNN algorithm on noisy data sets. For noise free data sets, the techniques may actually slightly improve the classification accuracy or at least do not degrade it. Other methods can be used to improve the classification accuracy of the kNN algorithm on noise free data sets such as building an ensemble of classifiers (6), (7).

This work is more elaborate and presents more weighting techniques than (5).

Khalil EL Hindi is with the CIS Department, University of Jordan (email: hindi@ ju.edu.jo).

Bayan Abu Shawar is with the CS Department, Arab Open University (email: bshawar@aou.edu.jo).

II. BAYESIAN-BASED WEIGHTING TECHNIQUES

In this work, we present 3 instance weighting techniques that are based on the conditional probability of the class of an instance given the instance's other attribute values.

A. Technique-1: The Conditional Probability of an Instance's Actual Class

This technique sets a weight to an instance that is equal to the conditional probability of its class given the values of the other attributes. That is

$$weight(inst) = p(c|val_1, val_2, \ldots val_n)$$

Where, *inst* is an instance, c is *inst*'s class and $val_1, val_2, \ldots val_n$ are *inst*'s values of attributes 1…n.

The $p(c|val_1, val_2, \ldots val_n)$ is computed using Naïve Bayesian rule, defined as

$$p(c|val_1, val_2, \ldots val_n) = \frac{p(val_1|c).p(val_2|c). \ldots .p(val_n|c).p(c)}{p(val_1, val_2, \ldots val_n)}$$

Where,

$$p(val_1, val_2, \dots, val_n) = p(val_1).p(val_2)\dots.p(val_n)$$

The intuition behind this technique is simple; if the probability of the instance's class given the instance's other attribute values is high, then that instance is probably a typical instance of that class, and should, therefore, be assigned a high weight, giving it a more influential vote. If, on the other hand, this probability is low, then the instance is probably untypical instance or even a noisy instance, and should, therefore, be assigned a small weight, giving it a lesser influential vote.

B. Technique-2: The Conditional Probability of the Class with the Second Highest probability

The second weighting technique, we present, is based on the following intuition. If there is a class, other than the instance's class, with a high conditional probability given the same instance attribute's values, then that instance is likely to be a noisy instance or at least untypical instance. In either case, its weight should be low. If, on the other hand, the probability of such a class is low, then that instance is probably a typical instance and should be given a large voting power.

That is the weight of the instance should be inversely proportional to probability that the instance belongs to another class. We define the weight according to this technique as

$$weight(inst) = 1 - p(c_o|val_1, val_2, \dots val_n)$$

Where c_0 is the class, other than the instance's actual class, with the highest probability, and $p(c_0|val_1, val_2, \dots val_n)$ is computed using the Naïve Bayesian rule described above.

C. Technique-3: The Difference Between the Probabilities that the Instance Belongs to its Actual Class or to the other Class with the Highest Probability

This technique, also, makes use of the class, other than the instance's actual class, with the highest probability, c_o. It takes the difference between the conditional probability of the instance's actual class and the probability of c_o given the instance attribute values. That is

$$diff(c, c_o) = p(c|val_1, val_2, \dots, val_n) - p(c_o|val_1, val_2, \dots, val_n)$$

It is easy to see that the higher this difference is the more voting power the instance should be given and the smaller this difference is, the smaller voting power the instance should be given. However, since this difference may be negative and to avoid giving negative weights to instances, we use the following equation to determine an instance's weight

$$weight(inst) = \max(0, diff(c, c_o))$$

III. Experimental Results

In this section, we discuss the empirical results we obtained using the weighting techniques discussed in the previous section. We used Weka's implementation of the kNN algorithm with a slight modification. We used the HVDM function (8) instead of the Hamming distance function to measure the distance between instances. The HVDM distance function we used is defined as

$$HVDM(New, Train) = \sqrt[2]{\sum_{a=1}^{m} d_a^2(New_a, Train_{haa})}$$

Where,

New and $Train$ are two inances

New_a and $Train_a$ are the values of attribute a in the instances New and Train, and

m is the number of attributes.

The definition of d_a is given below:

$$d_a(val_{new}, val_{train}) = \begin{cases} 1 & if\ val_{new}\ or\ val_{train}\ is\ unknown \\ diff_a(val_{new}, val_{train}) & if\ a\ is\ ordinal \\ vdm_a(val_{new}, val_{train}) & if\ a\ is\ nominal \end{cases}$$

We run the modified kNN algorithm without weighting and with weighting using each of the techniques discussed in the previous section. We used k=3 in all experiments. We used 39 data sets obtained from the UCI machine learning repository (9). We discretized the continuous attributes in the data sets, so that we can apply Naive Bayesian rule in a straightforward way. We used Fayyad et al.'s discretization method (10) as implemented in Weka to discretize the ordinal attributes. We used 10-fold cross validation in all our experiments.

To study the effectiveness of the technique in dealing with noisy data sets, we performed some experiments after inserting some noise in the training sets. The noise was inserted only in the training sets leaving the test sets untouched. In each experiment, we randomly selected a certain ratio of training instances and for each selected instance we flipped its class to some other class, also, randomly selected.

We experimented with different noise ratios of 0%, 10%, 20%, 30% and 40%. Due to the random nature of the selection process, we repeated the experiments, for each noise ratio (except of course for %0 noise), 5 times.

A. Results of Technique-1

Table 1 shows the average accuracy of each data set obtained using the kNN algorithm without weighting and with weighting using the first weighting technique. The experiment was repeated for 0% (no noise), 10%, 20%, 30% and 40% noise ratios. The last two rows of the table show the average accuracy over all data sets, and the number of data sets on which the corresponding technique achieved better results.

Fig. 1 shows the average accuracy graphically, while Fig. 2 shows the number of data sets on which the kNN achieves better results with and without the use of first weighting technique.

The results show that at %0 noise, using the weighting technique actually increased the average classification accuracy by 0.05% . Moreover, the number of data sets on which the weighting technique achieved better results is 13. While the number of data sets on which the kNN without weighting achieved better results is 9. The two methods achieved equal results on 18 data sets. This indicates that this weighting technique improves the classification accuracy even on noise free data sets.

Moreover, it can be easily seen that this weighting technique makes the kNN algorithm less sensitive to noise. This is evident from rate at which the classification accuracy

TABLE 1. THE AVERAGE ACCURACY, FOR EACH DATA SET, OBTAINED USING THE KNN WITH AND WITHOUT WEIGHTING, TECHNIQUE-1, AND FOR DIFFERENT NOISE RATIOS.

Data Set	Noise Ratio									
	0%		10%		20%		30%		40%	
	kNN	Tec1	kNN	Tec1	kNN	Tec1	kNN	Tec1	kNN	Tec1
sick	97.43	97.61	96.57	96.76	94.26	94.44	89.22	89.34	78.12	78.00
segment	95.80	95.63	93.71	94.85	88.44	92.87	80.73	89.02	70.03	82.76
lung-cancer	80.11	80.11	76.80	76.80	70.93	70.93	61.74	61.74	58.47	58.47
liver-disorders	54.84	54.84	55.93	54.94	58.28	57.99	59.84	59.79	56.38	55.39
labor	82.95	82.95	79.02	79.69	72.85	75.19	63.93	66.67	55.13	56.47
hepatitis	84.39	84.39	79.25	80.39	77.27	78.95	66.60	71.19	60.21	63.12
heart-h	79.17	79.86	77.65	79.43	78.87	80.04	76.55	77.66	70.69	72.98
heart-c	85.45	85.45	82.88	84.20	77.99	80.44	74.11	77.02	67.34	70.77
haberman	74.53	74.84	72.90	73.55	72.36	72.30	66.77	66.58	60.11	59.33
glass	77.22	79.10	75.91	76.96	74.31	75.64	70.28	72.73	64.35	66.90
flags	58.26	57.74	55.68	58.05	50.00	56.06	41.97	51.42	36.55	48.93
ecoli	76.31	76.31	76.79	76.84	77.02	76.13	75.20	74.91	74.80	75.58
diabetes	73.83	74.61	72.26	73.31	69.87	71.64	65.05	66.38	60.05	61.59
cylinder-bands	59.07	59.07	55.56	55.59	52.52	52.48	51.67	51.70	51.85	51.78
credit-g	72.30	72.30	68.38	68.38	65.12	65.12	60.08	60.06	55.40	55.38
credit-a	80.87	81.01	78.90	79.68	72.96	74.06	67.19	68.52	57.57	59.22
bridges_ver2	48.22	50.29	48.29	50.24	43.22	47.09	37.78	41.43	33.37	37.47
bridges_ver1	53.52	55.35	52.46	54.29	45.64	50.52	44.92	50.18	33.54	37.13
arrhythmia	74.34	70.81	73.76	70.14	71.43	68.55	67.08	65.18	61.47	61.25
audiology	71.82	69.57	68.45	67.67	63.95	64.90	60.34	62.12	50.74	54.41
breast-w	97.14	97.14	95.08	95.28	90.79	91.48	84.93	86.42	71.91	78.03
colic.orig	65.20	64.93	60.47	60.42	55.71	55.11	51.68	51.08	51.63	50.65
colic	85.05	85.32	80.46	80.68	74.42	74.91	69.03	69.14	57.69	57.64
autos	58.57	59.57	55.52	57.20	48.73	53.72	45.28	54.83	34.58	44.16
Car	90.39	90.39	88.14	87.97	84.68	84.52	76.97	76.78	68.48	68.83
breast-cancer	67.87	67.87	65.48	65.48	63.49	63.63	60.16	60.22	57.07	57.00
anneal	97.77	98.11	96.26	97.69	92.03	96.10	84.86	91.98	73.81	86.97
vote	95.39	95.39	93.28	93.60	87.48	89.54	79.02	83.34	63.77	69.64
vehicle	70.80	70.33	68.03	68.29	63.50	65.84	59.55	63.41	53.78	60.07
trains	60.00	60.00	60.00	60.00	56.00	56.00	54.00	54.00	58.00	58.00
vowel	62.22	62.12	59.94	61.27	56.61	59.62	52.63	57.23	47.60	54.14
nursery	88.12	87.73	86.57	87.79	81.65	85.41	74.11	81.39	64.43	75.56
mushroom	99.93	99.93	97.16	97.16	89.53	89.53	78.42	78.42	64.35	64.35
optdigits	97.38	97.38	95.11	96.71	89.28	94.89	81.44	91.93	70.96	87.19
pendigits	98.35	98.37	96.31	97.86	90.76	96.12	82.49	92.73	72.05	87.45
lymph	81.80	81.80	79.32	80.56	73.95	77.87	65.97	73.94	57.81	68.31
sonar	69.26	69.78	67.24	67.75	65.63	67.30	62.17	64.15	53.57	54.97
soybean	89.14	88.85	86.27	87.07	81.75	85.31	72.70	79.80	65.26	73.82
wine	98.33	98.33	96.42	97.11	89.28	93.47	81.69	86.99	68.72	78.68
average	78.29	78.34	76.11	76.71	72.12	73.99	66.62	69.78	59.27	63.65
#better	9	13	5	30	7	28	7	29	9	27

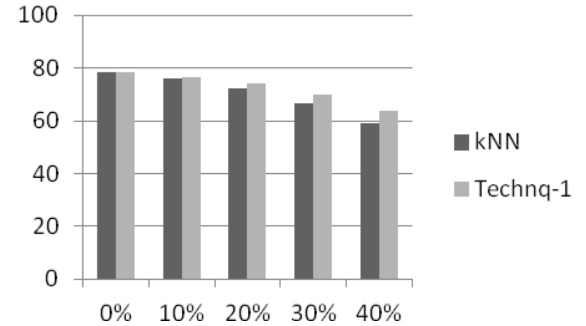

Fig. 1, the figure shows the average accuracy achieved by the kNN algorithm with and without using technique-1 and in the presence of different ratios of noise.

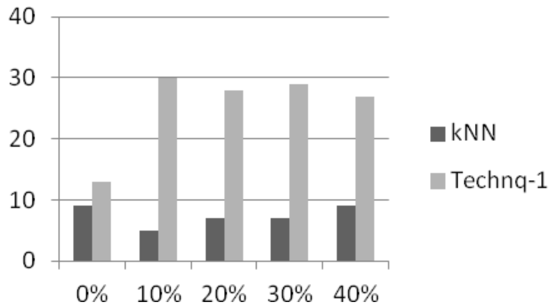

Fig. 2, the number of data sets on which the kNN with and without using technique-1 achieved better results, and in the presence of different noise ratios.

drops as we increase noise. It is, also, evident from the increase in the number of data sets the algorithm with weighting achieves better results, as the noise ratio increases.

The average accuracy of the kNN without instance weighting degraded from 78.29% at 0% noise to 76.11, 72,12, 66.62, and 59.27 at 10%, 20%, 30%, and 40% noise, respectively. On the other hand, the average accuracy of the kNN with the weighting technique, dropped at a much slower rate from 78.34% at 0% noise to 76.71, 73.99, 69.78, and 63.65 at 10%, 20%, 30%, and 40% noise.

While the average accuracy of the kNN without weighting dropped by 19.01 points as the noise ratio was increased from 0% to 40%, the average accuracy of the kNN with weighting dropped by only 14.69 points. Also, the number of data sets on which the kNN without weighting achieves better results quickly decreases as the noise ratio increases (see last row of table 1 and fig 2). Also, the number of data sets on which the algorithm with weighting achieved better results increased from 13 at 0% noise to 27 at 40% noise.

B. Results of Technique-2

A similar set of experiments were performed to test the effectiveness of the second weighting technique. Table 2 shows the average classification accuracy for the 39 data sets used.

At 0% noise this weighting technique achieves slightly better classification accuracy than the basic kNN (kNN without weighting). The average accuracy achieved using this weighting technique is 78.44% which makes its average accuracy 0.17% higher than the average accuracy of the basic kNN. Also, it achieves better results than the basic kNN on 15 data sets, worse results on 21 data sets and equal results on 3 data sets. It has, also, slightly better average classification accuracy than the first weighting technique on noise-free data sets. It, also, achieves better results on noisy data sets than the first weighting technique. Figure 3 and Figure 4 show the classification accuracy and the number of data sets on which kNN with and without weighting achieve better results, respectively. It is evident that the gap, in classification accuracy in favor of the weighting technique, gets wider as the noise ratio increases.

The average classification accuracy of this weighting technique dropped from 78.44% at 0% noise to 77.38%, 75.44%, 72.07 and 67.57%, at 10%, 20%, 30%, and 40% noise, respectively. At 40% noise the accuracy dropped by only 10.07% , while the accuracy of the basic kNN dropped by 18.33%. Moreover, this weighting technique achieves better results than the basic kNN on 15, 31, 33, 33, 34 data sets at 0%, 10%, 20%, 30%, and 40%noise. This is clear evidence that this weighting technique, actually, makes the kNN algorithm much less sensitive to noise. It is, also, better than the first weighting technique in this regard.

C. Results of Technique-3

In the third set of experiments, we tested the third weighting technique. Table-3 shows the average classification accuracy with and without weighting for the 39 data sets that we used. Figure 5 shows the classification accuracies for the kNN with and without weighting for different noise ratios. Figure 6 shows number of data sets at which the algorithm achieves better results in both cases. The results show that using this weighting technique makes the kNN more tolerant to noise and at the same time does not degrade its performance when used with noise free data sets. At 0% noise, the classification accuracies of the kNN with and without weighting are 78.29% and 78.27%, respectively. The number of data sets in which the kNN algorithm achieved better result is 13 in both cases (with and without weighting). These figures show that this weighting technique does not degrade the classification accuracy of the kNN even if the data sets are noise free. However, as the noise ratio increases the positive effect of this weighting technique becomes more evident both in the average classification accuracy and in the number of data sets on which the weighting techniques achieves better results. The results show that the classification accuracy of the kNN without weighting degrades from 78.29% at 0% noise to 76.04%, 72.37%, 66.16%, and 59.41% at 10%, 20%, 30%, and 40% noise, respectively. However, using the third weighting technique, the algorithm's performance degrades at a lower rate from 78.27% at 0% noise to 76.76%, 74.63%, 70.1%, and

TABLE 2. THE AVERAGE CLASSIFICATION ACCURACY ACHIEVED BY kNN WITH AND WITHOUT USING TECHNIQUE-2, AT DIFFERENT NOISE RATIOS.

Data Set	Noise Ratio									
	0%		10%		20%		30%		40%	
	kNN	Tech-2	kNN	Tech-2	kNN	Tech-2	kNN	Tech-2	kNN	Tech-2
Sick	97.43	97.32	96.59	96.75	94.32	95.21	88.96	91.26	78.18	81.39
Segment	95.83	94.98	93.62	94.43	88.52	92.58	80.59	90.16	70.87	85.51
lung-cancer	80.39	80.38	75.95	77.11	70.44	70.37	62.53	60.59	57.09	55.40
liver-disorders	55.29	55.29	56.48	55.18	58.09	57.40	57.36	57.61	58.13	57.93
Labor	82.46	82.46	81.04	82.72	75.18	76.22	62.89	69.08	55.99	60.23
Hepatitis	84.35	85.04	83.43	84.90	75.67	82.56	69.12	78.78	62.14	67.44
heart-h	79.26	82.58	78.85	81.66	78.74	81.06	76.63	79.88	72.88	77.23
heart-c	85.34	82.84	83.60	83.47	78.14	81.77	73.89	78.11	68.61	78.02
haberman	74.72	73.71	72.27	72.76	72.80	73.75	69.38	70.50	59.94	62.09
Glass	77.17	77.17	76.05	76.06	73.65	75.42	71.35	72.03	65.49	70.58
Flags	58.60	62.61	53.72	60.67	48.73	58.56	42.46	55.61	38.08	52.58
Ecoli	75.99	76.07	77.08	76.02	77.51	76.67	73.97	73.72	74.79	74.84
Diabetes	73.86	76.94	72.91	76.55	69.31	75.03	65.77	71.57	58.88	62.67
cylinder-bands	59.34	60.48	58.05	60.55	54.71	57.79	55.30	58.13	50.60	51.46
credit-g	72.07	73.66	68.43	71.76	65.13	70.27	60.25	65.45	55.51	58.14
credit-a	80.71	83.73	78.43	83.75	73.98	82.90	65.53	80.51	56.67	69.62
bridges_ver2	48.80	49.91	48.30	50.66	46.60	47.54	37.93	41.37	31.67	34.56
bridges_ver1	53.44	56.14	53.67	54.82	48.50	54.18	43.37	50.15	36.78	43.26
arrhythmia	73.97	61.65	72.72	59.70	70.01	59.08	66.62	57.88	60.90	56.71
audiology	71.86	66.39	68.42	64.15	64.45	56.74	57.25	53.12	52.08	51.21
breast-w	96.69	96.59	94.63	96.50	90.29	95.49	82.91	94.30	75.16	92.61
colic.orig	65.77	74.06	60.17	64.70	54.97	57.38	52.87	51.28	53.38	51.84
Colic	84.70	83.26	80.20	80.56	74.82	78.67	67.22	74.78	61.90	68.60
Autos	59.03	58.03	56.21	58.25	49.71	55.66	43.58	52.49	40.41	50.32
Car	89.83	88.45	88.19	88.63	82.99	84.47	77.50	79.45	69.41	71.22
breast-cancer	68.26	71.65	65.92	69.44	62.40	67.59	57.85	61.53	54.58	56.05
Anneal	97.25	97.64	95.32	97.31	91.76	96.91	83.41	94.57	73.62	89.59
Vote	95.42	94.53	93.23	93.31	88.52	92.36	77.34	89.28	63.40	87.07
vehicle	71.24	69.37	69.37	69.18	65.06	67.33	60.26	65.76	51.91	62.52
trains	60.20	60.17	60.17	60.16	60.09	60.13	58.04	58.14	55.93	56.12
vowel	62.19	63.18	60.36	62.26	55.64	58.49	53.60	56.63	47.33	52.17
nursery	87.25	86.83	85.39	86.58	80.62	85.20	73.66	82.18	63.59	76.37
mushroom	99.70	99.69	97.25	97.63	89.18	94.93	77.74	91.35	65.37	87.87
optdigits	97.43	96.99	95.20	96.51	89.15	95.40	80.92	93.07	70.53	89.08
pendigits	98.34	97.95	96.32	97.04	90.62	95.42	82.29	92.49	72.37	87.77
lymph	82.10	82.09	79.25	81.64	75.16	81.23	65.16	75.80	55.94	73.04
sonar	69.49	74.23	68.33	73.77	66.08	74.25	65.39	76.13	62.36	72.50
soybean	88.79	87.00	86.78	83.81	81.34	80.35	72.09	71.90	65.06	68.32
wine	98.16	98.13	93.96	96.87	89.51	95.89	81.73	94.02	70.26	91.16
average	78.27	78.44	76.30	77.38	72.37	75.44	66.53	72.07	59.94	67.57
#better	21	15	8	31	6	33	6	33	5	34

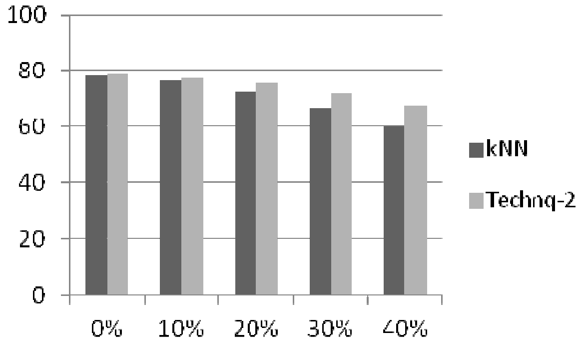

Fig. 3, the average classification accuracy of the kNN and without and with weighting using technique 2.

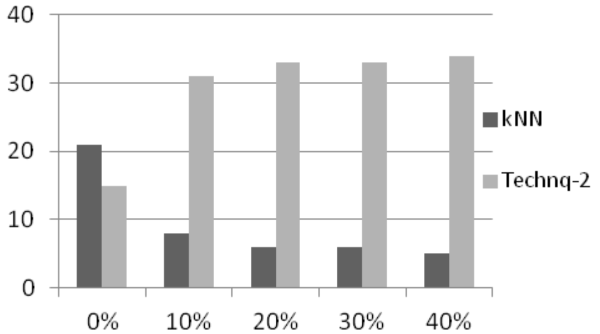

Fig. 4, the number of data sets on which the kNN achieves better results without and with weighting using technique 2.

64.74%, respectively. In other words, while the performance of the kNN dropped by 18.88% without weighting, its performance dropped by only 13.54% with this weighting technique. Also, the number of data sets at which this weighting technique achieved better results rose from 13 data sets at 0% noise to 26, 31, 28, and 28 data sets at 10%, 20%, 30%, and 40% noise, respectively.

The results show that although the three weighting techniques made the kNN more tolerant to noise, the second techniques proved to be more effective than the other two. Figure 7 shows the average classification accuracy of the basic kNN algorithm and each of the three weighting techniques at different noise ratios. The figure shows that the second weighting technique achieves better average classification accuracy than the kNN without weighting and with weighting using the first and third techniques and for different noise ratios.

IV. Conclusion

This work presented three Bayesian-based instance weighting techniques for the kNN algorithm that proved to make the algorithm more tolerant to noisy data set. The techniques are based on global information rather than local information. The first technique simply takes the conditional probability of the instance's actual class given the values of the instance's other attributes, as the weight of the instance. The second techniques is based on the conditional probability that the instance does not belong to a class other than its actual class. The third weighting technique simply takes the difference between of the two terms used in the first as second technique. Although, the all weighting techniques proved to make the kNN algorithm more tolerant to noise, the second weighting technique proved to be more effective than the other two techniques. This was evident in the average classification accuracy and in the number of data sets at which the second weighting techniques achieved better results than the other two. As a future work we intend to develop other weighting techniques and compare between them.

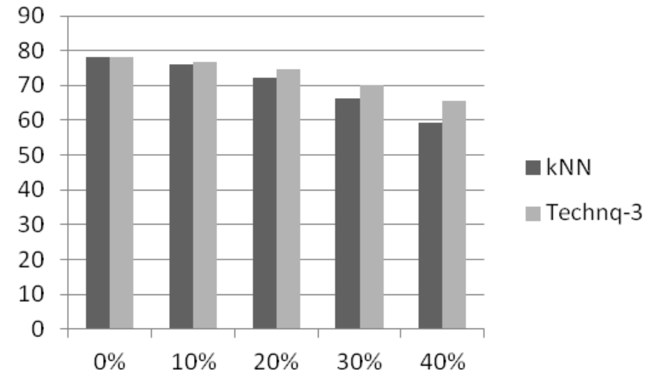

Fig. 5 The classification accuracy of the kNN algorithm with and without weighting using Technique-3

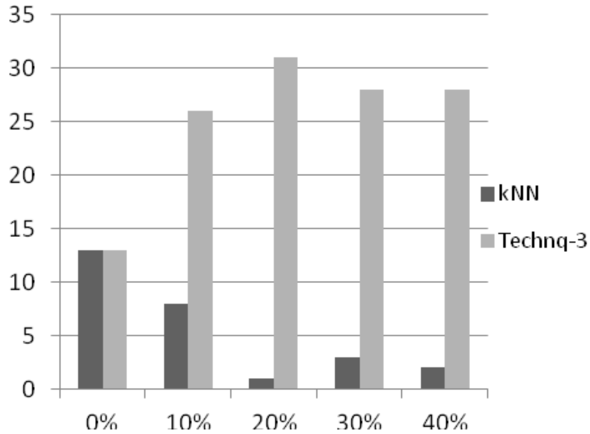

Fig 6. , the number of data sets on which the kNN achieved better results without and with weighting using technique 3.

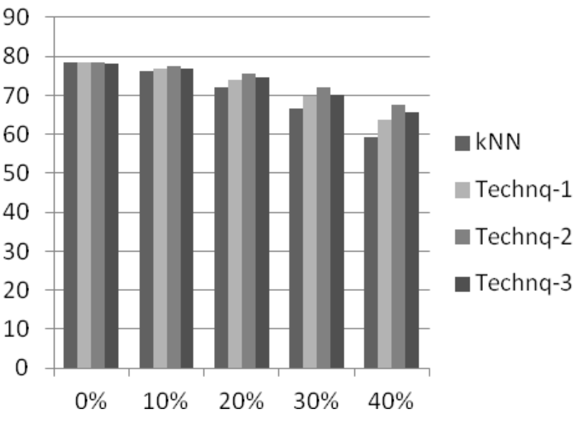

Fig. 7 The classification accuracy of the kNN algorithm without weighting and with weighting using each of techniques 1-3

TABLE 3 THE AVERAGE CLASSIFICATION ALGORITHM FOR EACH DATA SETS USING THEKNN ALGORITHM WITHOUT WEIGHTING AND WITH WEGHTING USING TECHNIQUE-3

Data Set	Noise Ratio									
	0%		10%		20%		30%		40%	
	kNN	Techq-3	kNN	Techq-3	kNN	Techq-3	kNN	Techq-3	kNN	Techq-3
Sick	97.43	97.03	96.69	96.53	94.15	95.40	89.57	91.65	77.82	80.42
Segment	95.80	94.76	93.64	94.23	88.46	93.23	80.89	91.03	70.15	86.82
lung-cancer	80.11	76.80	74.15	73.48	72.46	72.46	59.60	59.60	47.35	47.35
liver-disorders	54.84	54.84	55.20	55.20	57.53	57.53	56.76	56.76	54.63	54.63
Labor	82.95	82.95	81.15	78.93	71.11	71.96	62.65	66.85	56.74	60.88
Hepatitis	84.39	83.13	81.04	82.07	77.09	80.77	67.87	76.41	59.34	67.23
heart-h	79.17	81.85	78.35	81.59	76.97	81.26	75.12	79.30	72.98	78.26
heart-c	85.45	85.79	81.76	83.35	78.42	81.27	75.50	79.20	67.91	73.32
Haberman	74.53	75.16	73.55	73.94	70.26	70.45	67.74	67.99	59.42	59.42
Glass	77.22	79.10	75.73	76.88	73.48	75.27	69.73	70.98	65.12	66.35
Flags	58.26	58.26	52.62	52.71	46.50	46.88	41.98	42.39	35.32	35.84
Ecoli	76.31	76.60	78.22	76.64	77.06	75.85	74.79	73.51	73.59	73.47
Diabetes	73.83	75.13	72.31	74.16	68.83	70.75	65.49	68.38	59.42	61.27
cylinder-bands	59.07	59.07	57.19	57.19	55.04	55.04	53.78	53.78	50.22	50.22
credit-g	72.30	72.20	68.98	69.24	65.94	66.12	61.32	61.30	54.62	54.62
credit-a	80.87	82.46	77.88	81.59	74.72	79.94	66.03	70.87	57.25	59.01
bridges_ver2	48.22	48.22	47.29	47.67	41.66	42.06	38.77	39.17	31.40	31.40
bridges_ver1	53.52	53.52	51.87	52.61	46.38	46.76	39.47	39.47	33.37	33.55
Arrhythmia	74.34	74.34	73.20	73.20	70.53	70.53	67.97	67.97	62.44	62.44
Audiology	71.82	71.82	68.10	68.10	64.10	64.10	57.52	57.52	52.06	52.06
breast-w	97.14	97.14	94.71	96.48	91.76	96.00	84.53	93.60	73.14	84.61
colic.orig	65.20	67.13	58.75	59.71	57.21	58.42	54.43	54.00	52.27	53.14
Colic	85.05	80.43	82.59	80.16	75.64	78.41	68.52	74.18	61.03	67.17
Autos	58.57	58.07	56.16	56.77	51.54	55.02	44.34	47.84	40.69	46.31
Car	90.39	90.39	88.23	88.27	83.53	84.07	77.11	77.42	69.92	70.07
breast-cancer	67.87	67.87	65.50	65.64	60.78	60.78	60.06	60.06	54.17	54.10
Anneal	97.77	98.11	96.30	97.62	91.94	95.90	84.28	91.81	74.68	85.60
Vote	95.39	95.39	93.09	92.92	86.40	92.45	77.60	89.47	62.95	85.78
Vehicle	70.80	68.08	68.98	67.97	65.36	67.18	59.40	65.83	54.22	62.55
Trains	60.00	60.00	60.00	60.00	64.00	64.00	46.00	46.00	60.00	60.00
Vowel	62.22	62.53	59.82	60.57	57.41	59.27	52.24	54.61	46.83	49.13
Nursery	88.12	87.76	86.74	87.52	81.06	84.37	74.31	80.47	64.21	73.59
Mushroom	99.93	99.93	97.03	99.17	89.49	97.26	77.95	93.33	64.69	85.83
Optdigits	97.38	96.25	95.10	96.17	89.43	95.23	81.25	93.10	70.83	89.38
Pendigits	98.35	97.99	96.27	97.50	90.82	96.07	82.30	93.08	72.09	88.76
Lymph	81.80	83.14	77.20	80.59	73.76	78.63	68.79	74.72	55.89	62.54
Sonar	69.26	73.61	67.60	74.17	69.21	74.42	60.45	64.92	59.74	60.70
Soybean	89.14	87.97	87.09	86.62	80.50	80.94	71.52	73.89	66.21	69.67
Wine	98.33	97.78	95.32	96.55	91.82	94.66	82.42	91.43	72.13	87.20
Average	78.29	78.27	76.04	76.76	72.37	74.63	66.16	70.10	59.41	64.74
#better	1 3	1 3	8	2 6	1	3 1	3	2 8	2	2 8

REFERENCES

[1] **Aha David, Dennis Kibler, Marc Albert.** *Instance-Based Learning Algorithms*. 1991, Machine Learning, Vol. 6, pp. 37-66.

[2] ***IKONOMAKIS M., KOTSIANTIS S., TAMPAKAS V.,*** *Text Classification Using Machine Learning Techniques*. 8, 2005, WSEAS TRANSACTIONS on COMPUTERS, Vol. 4, pp. 966-974.

[3] ***KARAHOCA A., and NURULLAHOĐLU M.,*** *HUMAN MOTION ANALYSIS AND ACTION RECOGNITION*. Istanbul, Turkey : WSEAS, 2008. 1st WSEAS International Conference on Multivariate Analysis and its Application in Science and Engineering (MAASE '08).

[4] ***Wilson, D. R., and Martinez, T.R.*** *,Reduction Techniques for Exemplar-Based Learning Algorithms.* 2000, Machine Learning, Vol. 38, pp. 257-286.

[5] ***El Hindi K., and Abu Shawar B.,*** *Bayesian-Based Instance Weighting Techniques for Instance-Based.*Cambrige : WSEAS, 2012. International Conference on ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING and DATA BASES .

[6] ***Alkoot. F.,****Investigating methods for improving Bagged k-NN classifiers.* Cambridge, UK : s.n., 2008. 7th WSEAS Int. Conf. on ARTIFICIAL INTELLIGENCE, KNOWLEDGE ENGINEERING and DATA BASES.

[7] **ALKOOT F. AND QASEM** H.,,Experimental evaluation of diversity measures of combined classifiers.**.** AIKED-10, 2010 Cambridge, UK .

[8] ***Wilson D. R. and Martinez T. R*** *, Improved Heterogeneous distance functions.* 1997, Journal of Artificial Intelligence Research, Vol. 6, pp. 1-34.

[9] **Blake, C., and Merz, C.** s.l. : UCI Repository of Machine Learning Databases [online]. Irvine, CA: University of California, Department of Information and Computer Science. Available from: http://www.ics.uci.edu/~mlearn/MLRepository.html, 1998.

[10] ***Fayyad, U. Irani, K*** *Multi-Interval Discritization of Continuous Values Attributes for Classification Learning*: Morgan Kaufmann, 1993. Proceedings of 13th International Joint Conference on Artificial Intelligence. pp. 1022-1027.

[11] **Ian Witten, Eibe Frank,.** *Data Mining: Practical Machine Learning Tools and Techniques*. s.l. : Morgan Kaufmann, 2005.

[12] ***Aha, David.,*** *Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms.* 1992, International Journal of Man-Machine Studies, Vol. 36, pp. 267-287.