Advances in Knowledge-Based and Intelligent Information and Engineering Systems
M. Graña et al. (Eds.)
IOS Press, 2012

doi:10.3233/978-1-61499-105-2-59

Positive Predictive Value based dynamic *K*-Nearest Neighbor

Iñigo Mendialdua[1], Noelia Oses[2], Basilio Sierra[1] and Elena Lazkano[1]

[1] Department of Computer Science and Artificial Intelligence
University of Basque Country. http://www.sc.ehu.es/ccwrobot
[2] NOF Consulting. http://www.nofcon.com

Abstract. The K Nearest Neighbors classification method assigns to an unclassified observation the class which obtains the best results after a voting criteria is applied among the observation's K nearest, previously classified points. In a validation process the optimal K is selected for each database and all the cases are classified with this K value. However the optimal K for the database does not have to be the optimal K for all the points. In view of that, we propose a new version where the K value is selected dynamically. The new unclassified case is classified with different K values. And looking for each K how many votes has obtained the winning class, we select the class of the most reliable one. To calculate the reliability, we use the Positive Predictive Value (PPV) that we obtain after a validation process. The new algorithm is tested on several datasets and it is compared with the K-Nearest Neighbor rule.

Keywords: Nearest Neighbor, Supervised Classification, Machine Learning, Non-parametric Pattern Recognition

1 Introduction

In supervised classification problems [1] there are two extremes of knowledge which the modeler may consider. Either (s)he may have complete statistical knowledge of the underlying joint distribution of the observation x and the category θ, or (s)he may have no knowledge of the underlying distribution except that which can be inferred from samples. In the first extreme, a standard Bayes analysis will yield an optimal decision procedure and the corresponding minimum (Bayes) probability of error classification R^*. In the other extreme, a decision to classify x into the category θ is allowed to depend only on a collection of n correct samples $(x_1, \theta_1), (x_2, \theta_2), ..., (x_n, \theta_n)$, and the decision procedure is by no means clear. This problem is in the domain of supervised classification, and no optimal classification procedure exists with respect to all underlying statistics.

If it is assumed that the classified samples (x_i, θ_i) are independently identically distributed according to the distribution of (x, θ), certain heuristic arguments may be made about good decision procedures. For example, it is reasonable to assume that observations which are close together (in some appropriate distance metric) will have almost the same posterior probability distributions on their respective classifications.

Thus to classify the unknown sample x we may wish to weigh the evidence of the nearby x_i's most heavily. Perhaps the simplest non-parametric decision procedure of this form is the nearest neighbor (NN) classification method, which assigns to x the category of its nearest neighbor.

The first formulation of a rule of the NN type and primary previous contribution to the analysis of its properties is presumed to have been made by Fix and Hodges [2]. They investigated a method that is known as K Nearest Neighbors (K-NN), which assigns to an unclassified point the class most heavily represented among its k nearest neighbors.

In this paper we present a new version of K-NN method which finds the most reliable K for each new case. In a validation process we calculate the success rate when a class is predicted for different K values and for different amounts that belongs to the predicted class between these K Nearest Neighbors. Thus, when a new case has to be classified, we will select the K which get the most reliable result.

This paper is organized as follows. In section 2 we review the K-NN classification method while section 3 is devoted to related work in distance-based classifiers. The new proposed method is introduced in section 5. Section 6 shows the experimental results obtained and in the final section concluding remarks are presented.

2 The *K*-NN Classification Method

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be a correctly classified sample in classes $\theta_1, \ldots, \theta_M$, where $\mathbf{x}_i$ takes values in a metric space upon which a distance function d is defined. We will consider the pairs $(\mathbf{x}_i, \theta^i)$ where $\mathbf{x}_i$ is the p-variate observation upon the ith individual and θ^i is the class or category which that individual belongs to. We usually say that "$\mathbf{x}_i$ belongs to θ^i" when we mean precisely that the ith individual, upon which measurements $\mathbf{x}_i$ have been observed, belongs to category $\theta^i \in \{\theta_1, \ldots, \theta_M\}$.

Consider a new pair $(\mathbf{x}, \theta)$, where only the measurement $\mathbf{x}$ is observable, and where we estimate θ by using the information contained in the set of correctly classified points. We shall call $\mathbf{x}'$ the *nearest neighbor* (NN) of $\mathbf{x}$ if

$$\min_{i=1,\ldots,n} d(\mathbf{x}_i, \mathbf{x}) = d(\mathbf{x}', \mathbf{x}) \,. \tag{1}$$

The NN classification decision method gives to $\mathbf{x}$ the category θ^i, precisely the category of its nearest neighbor $\mathbf{x}_i$. In case of a tie between several neighbors, it has to be broken by modifying the decision rule.

An immediate extension to this decision rule is the so called K-NN approach [3], which assigns the candidate $\mathbf{x}$ the class which is most frequently represented among the k nearest neighbors of $\mathbf{x}$. In Figure 1, for example, the 3-NN decision rule would decide the class θ_o is active because two of the three nearest neighbors of $\mathbf{x}$ belong to class θ_o.

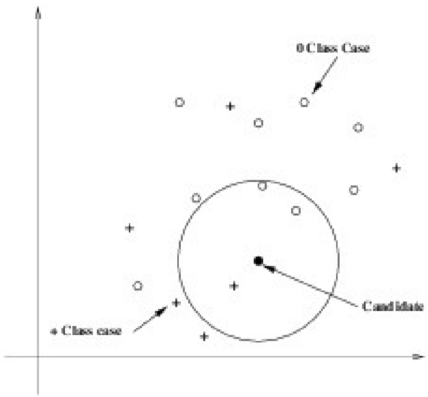

Fig. 1. Third Nearest Neighbor Decision Rule

3 Related work

Much research has been devoted to the K-NN rule [4]. One of the most important results is that K-NN has very good asymptotic performance. Broadly speaking, for a very large design set, the expected probability of incorrect classifications (error) R achievable with K-NN is bounded as follows:

$$R^* < R < 2R^* \tag{2}$$

where R^* is the optimal (minimal) error rate for the underlying distributions. This performance, however, is demonstrated for the training set size tending to infinity, and thus, it is not really applicable to real world problems in which we usually have a training set of hundreds or thousands cases, too few for the number of probability estimations to be performed.

Some distance based approaches, such that of Weinberger et al. [5] try to increase the obtained accuracy in distance based classification by looking for a specific distance, in an automatic way, for each classification problem. The proposed approach could be used to deal with unbalanced or biased databases; a similar idea can be found in other distance based methods [6]. On the other hand, *PEBLS* instance based inducer (Cost and Salzberg [7]) incorporates MVDM distance metric to deal with symbolic features, a modification of Stanfill and Waltz's VDM metric [8].

Improvement in classification can also be obtained by selecting and/or weighting features (see [9] for an example). Probabilistic voting approaches have also been used ([10], [11]); the main idea here is that each case among the K nearest ones make a weighted vote in favor of the class it belongs to, being the weight the probability each case has to belong to its own class. A very well known approach is the so called Instance Based Learning (IBL), based on the work of Aha [12] and Wettschereck [13]; there are several versions of the algorithm [14].

Another problem that arises with distance-based classifier systems is the management of large databases. Works devoted to data reduction [15] show interesting approaches which could also be used in combination with any other distance based algorithm when the size of the database is significant; other works in this field try to accelerate the execution of the distance based algorithm ([16], [17]) to obtain faster classifications.

Additionally, there are distance based classifiers which aim to deal with the so called multi labeling problem [18], in which, given a new case to be classified, a different number of categories could be given to it. For instance, and taking as example the document categorization area, a newspaper article relating the wedding of some country president would obtain Politics and Society as category labels, being both adequate for the document.

4 PPV *K*-NN

The standard procedure to use the K-NN algorithm with a new database corresponding to a classification problem, is to first select the appropriate K value for the datafile (K=5, for instance) and fix this value to deal with all the cases that have to be classified. The way the most suitable K value is obtained is mainly based on a validation process, although there are other possibilities found in the literature, for example Wang et al. [19] proposed a new method that dynamically adjust the number of nearest neighbors based on the statistical confidence.

As in Wang et al. work, in this paper we propose a new approach to determine the most likely value of K for each of the new cases to be classified. To do that, a validation approach is used as well, not to select a fixed K parameter for all the cases, but to select which K values are best for each of the different categories of the classification problem for the instance being processed.

To calculate which is the best K value for each new case, we use the Positive Predictive Value (PPV). The PPV is the proportion of instances which predicted to belong to certain class and are correctly classified. The Positive Predictive Value is defined as

$$PPV = \frac{numberofTruePositive}{numberofTruePositive + numberofFalsePositive} \tag{3}$$

where a True Positive is the event that the test makes a positive prediction, and the subject has a positive result. And a False Positive is the event that the test makes a positive prediction, and the subject has a negative result.

In a validation process, before the proper classifier is used, we calculate the PPV values for each predicted class, C_{Pr}, with different K values. As a result, for each C_{Pr} we obtain a PPV table which contains the different PPV values for each K value. Table 1 shows an example for a 3 class problem and K values from 1 to 4. We want to emphasize that the C_{Pr} is the class that a classifier predicted for the instance and no the class that the instance belongs to.

In order to make a deeper analysis of the classifier behavior and based on the results obtained for high values of K, we realize that it is not the same confidence

Table 1. Example of the PPV Tables, first version

C_{Pr}=1		C_{Pr}=2		C_{Pr}=3	
K	PPV	K	PPV	K	PPV
1	PPV_1	1	PPV_1	1	PPV_1
2	PPV_2	2	PPV_2	2	PPV_2
3	PPV_3	3	PPV_3	3	PPV_3
4	PPV_4	4	PPV_4	4	PPV_4

that 8 of the 9 neighbors belong to the C_{Pr} or to belong 5. This is the reason why we have extended the tables shown in Table 1 and take in consideration not only the K value and the C_{Pr}; we also take in consideration the amount of the cases that belongs to the C_{Pr} between the K neighbors, K_{Pr}. Hence, the tables that are used in classification task are extended to those shown in Table 2.

Table 2. Example of the PPV Tables, improved version

C_{Pr}=1			C_{Pr}=2			C_{Pr}=3		
K	K_{Pr}	PPV	K	K_{Pr}	PPV	K	K_{Pr}	PPV
1	1	PPV_1	1	1	PPV_1	1	1	PPV_1
2	2	PPV_2^2	2	2	PPV_2^2	2	2	PPV_2^2
	1	PPV_2^1		1	PPV_2^1		1	PPV_2^1
3	3	PPV_3^3	3	3	PPV_3^3	3	3	PPV_3^3
	2	PPV_3^2		2	PPV_3^2		2	PPV_3^2
	1	PPV_3^1		1	PPV_3^1		1	PPV_3^1
4	4	PPV_4^4	4	4	PPV_4^4	4	4	PPV_4^4
	3	PPV_4^3		3	PPV_4^3		3	PPV_4^3
	2	PPV_4^2		2	PPV_4^2		2	PPV_4^2
	1	PPV_4^1		1	PPV_4^1		1	PPV_4^1

Thus when a new case to be classified arrives, firstly we process it for different K values. For each K value, we get the C_{Pr} and the K_{Pr} and from the PPV tables we obtain the PPV value that correspond for each K. After that, we select the K with the highest PPV value and finally the K-NN result of this concrete K is assigned to the new case.

In Figure 2 we could see with an example how our algorithm works. In Figure 2(a) we could see the PPV tables that we obtain after a validation process. These tables show the real values that we got in our experiments with Ionosphere database. As it is a two-class problem there are two PPV tables, one for each C_{Pr}. On the other hand, in Figure 2(b) we could see the steps that our algorithm follows to classify a new case. In the first step, we get the 4 nearest neighbors classes. After that, in the second step, for different K values we apply the K-NN method obtaining the C_{Pr} and the K_{Pr}. In the third step, from the PPV tables we obtain the PPV value which belongs to the K, C_{Pr} and K_{Pr} values. And finally we select the case with the highest PPV value, and we assign to the new case its C_{Pr}.

C_{Pr}=0

	PPV	K_{Pr}	PPV	Numb. of Cases
K=1	0.8342	1	0.8342	199
K=2	0.8342	2	0.8865	185
		1	0.1429	14
K=3	0.8019	3	0.8956	182
		2	0.2333	30
K=4	0.8341	4	0.9133	173
		3	0.3784	37
		2	1	1

C_{Pr}=1

	PPV	K_{Pr}	PPV	Numb. of Cases
K=1	0.8689	1	0.8689	61
K=2	0.8689	2	0.9189	37
		1	0.7917	24
K=3	0.9167	3	0.96	25
		2	0.8696	23
K=4	0.9184	4	0.9474	19
		3	0.8947	19
		2	0.9091	11

(a) PPV table for each C_{Pr}

[0,1,1,0]

1. 4–NN Classes

K	C_{Pr}	K_{Pr}
1	0	1
2	0	1
3	1	2
4	0	2

2. Get C_{Pr} and K_{Pr} for each K

K	C_{Pr}	K_{Pr}	PPV
1	0	1	0.834
2	0	1	0.142
3	1	2	0.869
4	0	2	1*0.834=0.834

3. Get PPV value from PPV tables

$K = 3$

PPV = 0.869

Final Class = 1

4. Select the highest PPV

(b) Step by step

Fig. 2. In 2(a) we show the tables that our algorithm get in our experiments after a validation process with Ionosphere database. Each C_{Pr} table shows the PPV for different K, with different K_{Pr}. In 2(b) we show the steps that our algorithm follows to assign the final class

Sometimes it is possible to be few cases to achieve the PPV value. For example, in Figure 2(a), in $C_{Pr} = 0$ Table, we could see that when $K = 4$ and $K_{Pr} = 2$, there is only one case. As this case has been classified correctly in the validation process, its success rate is very high. This success rate is not very reliable because they do not have enough cases to support this claim, and it is possible that the case correctly classified in the validation process could be an exception. To avoid these cases we have included a threshold. If there is no more than 10 cases, the degree of confidence is multiplied by the degree of confidence of the K. We could see that in Figure 2(b) for $K = 4$ case.

5 Experimental Results

In this section we show the experimental results obtained with different databases. We have compared our method with K-NN for different K values.

5.1 Datasets

Twenty-six databases are used to test our hypothesis. All of them are obtained from the *UCI Machine Learning Repository* [20]. The characteristics of the databases are given in Table 3.

Table 3. The characteristics of the 26 databases used in this experiment

Domain	Num. of Instances	Num. of Attributes	Num. of Classes
Australian Credit	690	14	2
Balance	625	4	3
Blood Transfusion	748	5	2
Breast Cancer	569	32	2
Car	1728	6	4
Cmc	1473	9	3
Diabetes	768	8	2
Glass	210	9	7
Haberman	306	3	2
Image Segmentation	2310	19	7
Ionosphere	351	34	2
Iris	150	4	3
Letters	20000	16	26
Magic Gamma Telescope	19020	11	2
Optical Digits	5620	64	10
Pen-Based Digits	10992	16	10
Sonar	208	60	2
Spambase	4601	57	2
Statlog(German)	1000	20	2
Statlog (Heart)	270	13	2
Statlog (Landsat)	6435	36	7
Statlog (Shuttle)	58000	9	7
Tic Tac Toe	958	9	2
Vowel Context	528	10	11
Waveform	5000	21	3
Wine	178	13	3

5.2 Experimental setup

In order to give a real perspective we have applied 5x2 fold cross validation to each database [21]. But firstly our algorithm needs a validation process to get the PPV values. So we have applied 5-hold out for each fold, where we have used the %70 as training and %30 as testing.

In Table 4 we show the results obtained by the K-NN method for different K values, where the best result is written in boldface. We do not include the K=2 value in this Table, because in the case of draw we give the preference to the nearest neighbor, and hence the K=2 and K=1 results always are the same.

In Table 5 we compare our method with the K-NN. Looking at the results of Table 5, it could be seen that in most of the cases our method improves

Table 4. Accuracy level percentage of the K-NN method

Database	K-NN=1	K-NN=3	K-NN=4	K-NN=5	K-NN=6	K-NN=7	K-NN=8	K-NN=9	Avg	σ
Australian Credit	80.203	83.275	83.565	84.812	84.783	84.870	84.957	85.072	83.942	1.6561
Balance	78.814	81.571	83.109	84.808	86.571	87.436	87.660	87.917	84.736	3.3180
Blood Transfusion	68.797	73.396	73.743	75.374	75.802	76.337	76.417	77.406	74.659	2.7275
Breast Cancer	95.458	96.725	96.761	96.444	96.725	96.444	96.585	96.232	96.422	0.4296
Car	85.509	90.660	90.880	91.551	91.123	90.949	90.856	90.729	90.282	1.9483
Cmc	43.505	45.516	46.250	46.685	46.766	47.242	46.834	46.861	46.208	1.2078
Diabetes	69.474	74.323	73.308	75.602	74.023	75.489	75.451	75.789	74.182	2.1008
Glass	64.299	65.421	63.271	64.860	63.084	63.645	62.523	63.178	63.785	0.9878
Haberman	65.098	68.039	67.908	69.281	69.412	71.830	71.699	73.203	69.559	2.6199
Image Segmentation	95.792	94.909	94.814	94.035	94.424	93.844	94.147	93.706	94.459	0.6897
Ionosphere	85.486	84.171	84.400	83.771	84.171	83.600	83.600	82.743	83.993	0.7909
Iris	93.867	94.400	94.400	94.933	94.267	94.267	94.267	94.933	94.417	0.3595
Letters	94.343	94.291	94.422	94.016	94.098	93.647	93.609	93.217	93.955	0.4251
Magic Gamma Telescope	80.124	82.211	82.462	82.980	83.115	83.212	83.281	83.213	82.575	1.0636
Optical Digits	98.359	98.456	98.491	98.349	98.406	98.185	98.256	98.139	98.330	0.1263
Pen-Based Digits	99.212	99.128	99.148	98.970	99.034	98.810	98.866	98.675	98.981	0.1859
Sonar	82.788	77.692	79.423	74.519	75.385	70.192	72.019	68.173	75.024	4.8764
Spambase	88.530	88.965	89.583	88.757	89.435	88.596	89.178	88.130	88.897	0.4893
Statlog (German)	66.960	69.200	68.840	70.000	70.520	70.620	71.580	71.000	69.840	1.4723
Statlog (Landsat)	89.473	90.002	89.983	89.890	90.045	89.557	89.765	89.442	89.770	0.2478
Statlog (Heart)	75.778	78.963	78.889	81.037	80.222	80.370	80.519	80.593	79.546	1.7060
Statlog (Shuttle)	99.932	99.876	99.880	99.835	99.835	99.794	99.792	99.751	99.837	0.0581
Tic-tac-toe	100.000	99.311	99.478	97.745	98.455	96.013	97.641	95.177	97.978	1.6995
VowelContext	93.758	80.202	76.687	63.737	59.758	49.394	44.626	38.646	63.351	19.1055
Waveform	77.160	80.448	80.348	81.748	81.476	82.696	82.504	83.044	81.178	1.9020
Wine	94.719	95.506	95.843	95.506	95.730	96.180	96.292	96.742	95.815	0.6117

the mean of all the K, only getting worse results in 5 databases. This can be considered logical because in Table 4, it can be seen that in some databases there are considerable differences for different values of K, which makes to devalue the mean.

On the other hand, if we compare our method with the best K, we could see that it improves in 6 databases, it draws in 1 and get worse in 20. Although this results do not seem promising, looking in more detail, we could see that the majority of the improved databases have low standard deviation value, σ, for the different K values. There are 12 databases which has $\sigma < 1$, and the results are very similar; in 5 of them our method gets the best result, in 1 they draw and in the other 6 the best K-NN is the optimal. We think that these are optimistic results taking into account that we are comparing our methods only result with the best between the K-NN's 9 results.

6 Conclusion and Further Results

A new method extending the K-NN idea is presented in this work: PPV K-NN. The main reason of this approach is to ensure that the final decision is made with some confidence. For doing that we search the most reliable K for each new case.

The new method has been implemented and tested over 26 databases from the UCI repository. We have compared our method with the best result and the mean of K-NN. Getting interesting results for the databases with regular results for different K in K-NN method.

Table 5. Comparisson between our method resultd with K-NN's best result and mean

Database	PPV K-NN	K-NN Best	Avg K-NN
Australian Credit	84.116	**85.072**	83.942
Balance	86.635	**87.917**	84.736
Blood Transfusion	76.364	**77.406**	74.659
Breast Cancer	96.655	**96.761**	96.422
Car	90.197	**91.551**	90.282
Cmc	**47.541**	47.242	46.208
Diabetes	73.947	**75.789**	74.182
Glass	64.766	**65.421**	63.785
Haberman	72.484	**73.203**	69.559
Image Segmentation	95.446	**95.792**	94.459
Ionosphere	**88.286**	85.486	83.993
Iris	94.133	**94.933**	94.417
Letters	**94.659**	94.422	93.955
Magic Gamma Telescope	82.387	**83.281**	82.575
Optical Digits	**98.491**	**98.491**	98.330
Pen-Based Digits	**99.216**	99.212	98.981
Sonar	79.904	**82.788**	75.024
Spambase	89.561	**89.583**	88.897
Statlog (German)	70.700	**71.580**	69.840
Statlog (Heart)	79.778	**81.037**	79.546
Statlog (Landsat)	**90.228**	90.045	89.770
Statlog (Shuttle)	99.924	**99.932**	99.837
Tic-tac-toe	99.937	**100.000**	97.978
VowelContext	92.667	**93.758**	63.351
Waveform	80.720	**83.044**	81.178
Wine	**96.966**	96.742	95.815

As future work it would be interesting to include the reject option. Where we reject the cases where there are two different K with a high PPV value that predict different classes.

References

1. Mitchell T.: Machine Learning. McGraw-Hill (1997)
2. Fix, E., Hodges Jr, J.L.: Discriminatory analysis, nonparametric discrimination. Technical Report Project 21-49-004, USAF school of Aviation Medicine, Randolf field (1951)
3. Cover, T.M., Hart, P.E.: Nearest Neighbor Pattern Classification. IEEE Trans. IT-13, vol. 1, pp. 21-27 (1967)
4. Dasarathy, B.V.: Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques. IEEE Computer Society Press (1991)
5. Weinberger, K.Q., Saul, L.K.: Distance Metric Learning for Large Margin Nearest Neighbor Classification. The Journal of Machine Learning Research. vol. 10, pp. 207-244 (2009)
6. Tan, S.: Neighbor-Weighted K-Nearest Neighbor for Unbalanced Text Corpus. Expert Systems with Applications. vol. 28, pp. 667-671 (2005)
7. Cost, S., Salzberg, S.: A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning, vol. 10, num. 1 , pp. 57-78 (1993)
8. Stanfill, C., Waltz, D.: Toward Memory-Based Reasoning. Communications of the ACM, vol. 29, num. 12, pp. 1213-1228 (1986)
9. Tahir, M.A., Bouridane, A., Kurugollu, F.: Simultaneous Feature Selection and Feature Weighting Using Hybrid Tabu Search/K-Nearest Neighbor Classifier. Pattern Recognition Letters archive, vol. 28, pp. 438-446 (2007)

10. Martnez-Otzeta, J.M., Sierra, B.: Analysis of the Iterated Probabilistic Weighted k-Nearest Neighbor Method, a New Distance-Based Algorithm. 6th International Conference on Enterprise Information Systems (ICEIS), vol. 2, pp. 233-240 (2004)
11. Sierra, B., and Lazkano, E.: Probabilistic-Weighted k Nearest Neighbor Algorithm: a New Approach for Gene Expression Based Classification. KES02 proceedings (IOS press), pp. 932-939 (2002)
12. Aha, D.: Tolerating, Irrelevant and Novel Attributes in Instance-Based Learning Algorithms. International Journal of Man-Machine Studies, vol. 36, pp. 267-287 (1992)
13. Wettschereck, D.: A Study of Distance-Based Machine Learning Algorithms. Ph.D. Thesis, Oregon State University (1994)
14. Aha, D., Kibler, D., Albert, M.K.: Instance-Based learning algorithms. Machine Learning, vol. 6, pp. 37-66 (1991)
15. Fayed, H.A., Atiya, A.F.: A Novel Template Reduction Approach for the K-Nearest Neighbor Method. IEEE Transactions on Neural Networks, vol. 20, num. 5, pp. 890-896 (2009)
16. Wang, Q., Kulkarni, S.R., Verdu, S.: Divergence Estimation for Multidimensional Densities Via K-Nearest-Neighbor Distances. IEEE Transactions on Information Theory, vol. 55, pp. 2392-2405 (2009)
17. Liaw, Y.C., Leou, M.L., Wu, C.M.: Fast Exact K Nearest Neighbors Search Using an Orthogonal Search Tree. Pattern Recognition, vol. 43, pp. 2351-2358 (2010)
18. Younes, Z., Abdallah, F., Denux, T.: Multi-Label Classification Algorithm Derived from K-Nearest Neighbor Rule with Label Dependencies. 16th European Signal Processing Conference (2008) , C., Boto, F., Soraluze, I., Perez, A.: An incremental and hierarchical k-NN classifier for handwritten characters. In: Proc. 16th Internat. Conf. on Pattern Recognition, Quebec City, Canada, vol. 3, pp. 98-101 (2002)
19. Wang, J., Neskovic, P., Cooper, L.N.: Neighborhood selection in the k-nearest neighbor rule using statistical confidence. Pattern Recognition, vol. 39 pp. 417423 (2006)
20. Frank, A., Asuncion, A.: UCI Machine Learning Repository, http://archive.ics.uci.edu/ml, Irvine, CA: University of California, School of Information and Computer Science (2010)
21. Demsar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. Journal of Machine Learning Research, vol. 7, pp. 1-30 (2006)