Weighted Instance-Based Learning Using Representative Intervals

Octavio Gómez, Eduardo F. Morales, and Jesús A. González

National Institute of Astrophysics, Optics and Electronics,
Computer Science Department,
Luis Enrique Erro 1, 72840 Tonantzintla, México
{gomezo,emorales,jagonzalez}@ccc.inaoep.mx
http://ccc.inaoep.mx

Abstract. Instance-based learning algorithms are widely used due to their capacity to approximate complex target functions; however, the performance of this kind of algorithms degrades significantly in the presence of irrelevant features. This paper introduces a new noise tolerant instance-based learning algorithm, called WIB-*K*, that uses one or more weights, per feature per class, to classify integer-valued databases. A set of intervals that represent the rank of values of all the features is automatically created for each class, and the nonrepresentative intervals are discarded. The remaining intervals (representative intervals) of each feature are compared against the representative intervals of the same feature in the other classes to assign a weight. The weight represents the discriminative power of the interval, and is used in the similarity function to improve the classification accuracy. The algorithm was tested on several datasets, and compared against other representative machine learning algorithms showing very competitive results.

Key words: Feature Weighting, Instance-Based Learning, *K*-NN

1 Introduction

Instance-based learning algorithms are derived from the nearest neighbor pattern classifier [6], and their design was also inspired by exemplar-based models of categorization [17]. Unlike other learning methods that construct an explicit description of the target function from the training examples, instance-based learning algorithms only store the examples, and delay the processing effort until a new instance need to be classified. Instance-based learning algorithms are widely used due to their advantages that include small training cost, efficiency gain through solution reuse [1], high capacity to model complex target functions and their ability to describe probabilistic concepts [2]. The performance of this kind of algorithms, however, degrades significantly in the presence of irrelevant features; so, distinguishing relevant features is a very important issue.

One way to improve the robustness of instance-based learning algorithms against irrelevant features is through feature weighting. In feature weighting, each feature is multiplied by a weight value proportional to its ability to distinguish among classes. There are many algorithms of feature weighting. M. Tahir et al. (2007) [18] proposed a hybrid approach to simultaneously perform feature selection and feature weighting based on tabu search (TS) and the K-NN algorithms; they modified the solution encoding used by the TS algorithm by adding feature weights and binary feature vectors, and then used a K-NN classifier to evaluate the sets of weights produced by tabu search. Blansch et al. (2006) [4] proposed a method that performs a modular grouping of complex data called MACLAW. This method assigns weights to the features under a wrapper approach. A set of extractors is defined, and all the extractors are associated to a clustering algorithm and a local weights vector. The weights are obtained from standard cluster quality measures such as compactness. De la Torre et al. (2002) [19] applied the discriminative feature extraction (DFE) method to weight the contribution of each component of a feature vector. The weights are obtained from the partial probability weighting (PPW) exponents, and each weight represents the partial probability of each component of the feature vector. Thomas Gartner and Peter A. Flach (2000) [10] proposed an algorithm that combines naive bayes classification with feature weighting. They employ a support vector machine to weight the features, and, the weights are optimized to reduce the danger of overfitting. K. Kira and L. Rendell (1992) [12] proposed a weighting algorithm called RELIEF that estimates a feature weight $W[A]$ as an approximation of the difference of probabilities P(different value of A | nearest instance of different class) - P(different value of A | nearest instance of the same class) where A is an attribute. This algorithm was designed for 2-class problems. I. Kononenko (1994) [13] extended the RELIEF algorithm to deal with multi class problems, finding the probabilities with respect to each class and averaging their contribution. A good review and empirical evaluation of many feature weighting methods can be found in (Wettschereck et al., 1997) [21].

This paper introduces a new instance-based learning algorithm, called WIB-K, that uses one or more weights, per feature per class, for the classification of integer-valued databases, and that is noise tolerant. A set of intervals that represents the rank of values of all the features is automatically created for each class, the representative intervals are located by means of a majority criterion, and the nonrepresentative intervals are considered as noise (outliers). The representative intervals of each feature are compared against the representative intervals of the same feature in the other classes to obtain a weight; this weight represents the discriminative power of the interval, and is used in the similarity function to improve the classification rate. The proposed algorithm was tested on several integer-valued datasets from the UCI repository, and compared against other representative machine learning algorithms, showing very competitive results.

The paper is organized as follows. Section 2 gives an overview of instance-based learning. In Section 3 the weighting schema is described. In Section 4 the experimental results are presented and, in Section 5, the main conclusions and a brief discussion of future work is given.

2 Instance-Based Learning

2.1 Learning Task and Framework

Instance-based learning algorithms are derived from the nearest neighbor pattern classifier [6]. This kind of algorithms stores and uses only selected instances to generate classification predictions by means of a distance function. The learning task of these algorithms is supervised learning from examples.

Each instance is represented by a set of attribute-value pairs, and all instances are described by the same set of n attributes. This set of n attributes defines an n-dimensional instance space. One of the attributes must be the category attribute and the other attributes are predictor attributes.

The primary output of an instance-based learning algorithm is a function, that maps instances to categories, called concept description; this concept description includes a set of stored instances and, possibly, information about the classifier past performance. The set of stored instances can be modified after each training instance is processed. All instance-based learning algorithms are described by the following three characteristics:

1. *Similarity function:* computes the similarity between a training instance i and the instances stored in the concept description. The similarities are numerical-valued.
2. *Classification function:* This function receives the results of the similarity function and the performance records stored in the concept description. It yields to a classification for the training instance i.
3. *Concept description updater:* Keeps the records of classification performance and decides the instances to be included in the concept description. It yields to a modified concept description.

The similarity and classification functions determine how the instances stored in the concept description are used to predict the category of the training instance i.

2.2 IB-K Algorithm

IB-K is a very straightforward instance-based learning algorithm. The distance function that it uses is:

$$Distance(x, y) = \sqrt{\sum_{i=1}^{n} f(x_i, y_i)}$$

where x is a test instance, y is a training instance, x_i is the value of the i-th attribute of instance x and $f(x, y)$ is defined as follows:

$$f(x_i, y_i) = (x_i - y_i)^2$$

Table 1. IB-K algorithm (CD = concept description)

```
$CD \leftarrow$ *all the labeled instances*
**For each** $x \in$ Training Set **do**
  1. **For each** $y \in CD$ **do**
     $Dist \leftarrow Distance(x, y)$
     **If** $Dist$ **is one of the K-smallest distances** $Ksmall[m] \leftarrow Dist$
  2. $class(x)$ = majority class present in $Ksmall[m]$
  3. $CD \leftarrow CD \cup x$
```

The instances are described by n features. The IB-K algorithm is presented in Table 1.

In order to label an instance, the IB-K algorithm computes the distance between the test instance and the instances stored in the concept decription, and stores the K nearest instances. The class of the test instance will be the preponderant class of the K nearest instances previously obtained.

3 Feature Weighting Based on Representative Intervals

3.1 Initial Definitions

- Ω is the instance space formed by n instances and m features.
- $x_i \in \Omega$ represents the i-th instance, $1 \leq i \leq n$.
- $x_{i,j}$ represents the value of the j-th feature of the i-th instance, $1 \leq j \leq m$.
- C^α_β is a multiset that contains all the values of feature β for all the instances $x_i \in \Omega$ with $class(x_i) = \alpha$.

$$C^\alpha_\beta = \{x_{i,j} | class(x_i) = \alpha \wedge j = \beta\}$$

- D^α_β is the set that contains all the values contained in C^α_β, but without repeated values. This set is partially ordered under the function $<$. For example, if $C^\alpha_\beta = \{3, 5, 7, 4, 9, 3, 9, 5, 3, 5, 4, 6\}$ then $D^\alpha_\beta = \{3, 4, 5, 6, 7, 9\}$.
- $f(a)$ is the frecuency function, it returns the number of times that a value $a \in D^\alpha_\beta$ appears in C^α_β, and is defined as follows

$$f(a) = \sum_{\forall b_l \in C^\alpha_\beta} g(a, b_l)$$

where $1 \leq l \leq |C^\alpha_\beta|$ and $g(a, b_l)$ is defined as

$$g(a, b_l) = \begin{cases} 1 \text{ if } a = b_l \\ 0 \text{ otherwise} \end{cases}$$

For example, with the previous sets C^α_β and D^α_β, $f(5) = 3$. This function can be viewed as histogram of the image.

3.2 Representative Intervals and Weights

First, the D^{α}_{β} set must be partitioned into collectively exhaustive and mutually exclusive subsets $D^{\alpha}_{\beta,\gamma}$, where γ is the index of the partition. All the consecutive intervals must be grouped in exactly one partition. For example, if $D^{\alpha}_{\beta} = \{1, 2, 3, 5, 6, 7\}$ then the only resultant partitions are $D^{\alpha}_{\beta,1} = \{1, 2, 3\}$ and $D^{\alpha}_{\beta,2} = \{5, 6, 7\}$.

The magnitude of a partition $D^{\alpha}_{\beta,\gamma}$ is:

$$Magnitude(D^{\alpha}_{\beta,\gamma}) = \sum_{\forall t \in D^{\alpha}_{\beta,\gamma}} f(t)$$

The amplitude of a partition $D^{\alpha}_{\beta,\gamma}$ is:

$$Amplitude(D^{\alpha}_{\beta,\gamma}) = |D^{\alpha}_{\beta,\gamma}|$$

All the partitions $D^{\alpha}_{\beta,\gamma}$ are grouped according to their amplitude. Let $E^{\alpha}_{\beta,\eta}$ be the set formed by all the partitions $D^{\alpha}_{\beta,\gamma}$ with the same amplitude η, then, the maximum amplitude ψ of the set $E^{\alpha}_{\beta,\eta}$ is:

$$\psi = argmax(Magnitude(D^{\alpha}_{\beta,\gamma}))$$

where $D^{\alpha}_{\beta,\gamma} \in E^{\alpha}_{\beta,\eta}$. In order to discard the nonrepresentative intervals, considered as noise (outliers), it is necessary to define levels of confidence. This characteristic allows the algorithm to be noise-tolerant.

If ψ is the maximum amplitude of $E^{\alpha}_{\beta,\eta}$ then the levels of confidence are shown in Table 2, where % represents the integer division.

Table 2. The four levels of confidence defined to discriminate noise

Level of confidence	Interval	Left value	Right value
High	$[H_i, H_f]$	$H_i = (H_f\%2) + 1$	$H_f = \psi$
Medium	$[M_i, M_f]$	$M_i = (M_f\%2) + 1$	$H_i - 1$
Low	$[L_i, L_f]$	$L_i = (L_f\%2) + 1$	$M_i - 1$
Null	$[0, N_f]$	0	$L_i - 1$

The sets $D^{\alpha}_{\beta,\gamma} \in E^{\alpha}_{\beta,\eta}$ which magnitude falls in the null level of confidence are discarded because they are considered noise (outliers). The remaining sets $D^{\alpha}_{\beta,\gamma}$ are the representative intervals of feature β for class α.

The percentage of values inside a representative interval of a feature β that are not overlapped with any other value inside all the representative intervals of the same feature β for all the remaining classes is the weight of the interval. The weight must be in the range $[0, 1]$. For example, if an interval of 30 values has 10 overlapped values, its weight is $(30 - 10)/30$. The different types of overlap between two intervals are shown in Fig. 1. In general, a given interval is overlapped by combinations of these base overlaps.

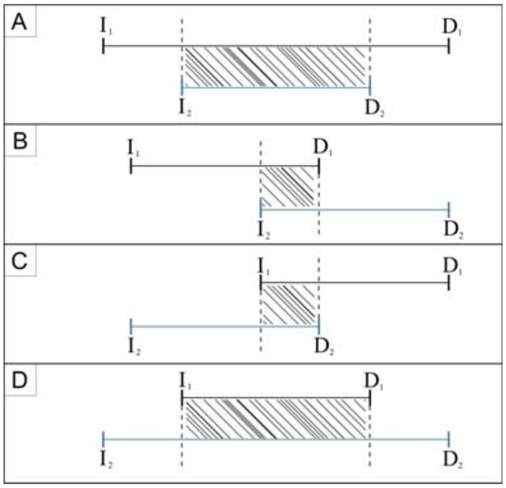

Fig. 1. The four types of overlap between two intervals: totally overlapped(A) where *non-overlapped area* $= 0$, partially left-overlapped (B) where *non-overlapped area* $= (D_2 - I_2) - (D_1 - I_2)$, partially right-overlapped (C) where *non-overlapped area* $= (D_2 - I_2) - (D_2 - I_1)$ and partially center-overlapped (D) where *non-overlapped area* $= (D_2 - I_2) - (D_1 - I_1)$

The obtained weights are used in the distance function of WIB-k:

$$Distance(x, y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2 w(y_i)}$$

where x is the example to label and y is the labeled example stored in the concept description of WIB-K. If y_i falls within a representative interval, its weight will be the weight of the interval. If $x_{i,j}$ does not fall in any representative interval, its weight will be the weight of the closest representative interval. The weights are normalized.

4 Results

4.1 Data Sets

We performed experiments and comparisons over several real world datasets from the UCI machine learning repository [14] in order to demonstrate the performance of the proposed algorithm. We selected databases with integer-valued features, without concerning about the type of the class. A brief description of the datasets is given in Table 3.

All the data sets have been randomly partitioned in ten disjoint sets for 10-fold cross validation. The same training and testing sets were used for all the algorithms. Instances with missing values were removed. The compared algorithms were taken from Weka class library [8] and the parameters used are the default parameters, except for the K value that always was the same value used in WIB-K.

Table 3. Description of the eight data sets used for experiments and comparisons

Name	Instances	Features	Classes
Balance Scale (BS)	625	4	3
Breast Cancer (BC)	699	11	2
CMC	1473	10	3
Dermatology (D)	366	34	6
Haberman (H)	306	4	2
Hayes Roth (HR)	162	6	3
Lung Cancer (LC)	32	57	3
TAE	151	6	3

4.2 Comparison Against Instance-Based and Weighted Instance-Based Algorithms

This subsection shows the results of the comparison of W-IBK against other weighted and non weighted instance-based learning algorithms. IB1 and IB-K are the implementations of the original instance-based learning algorithms proposed by D. Aha et al. [2]. dw-IBK(1/d) and dw-IBK(1-d) are the IB-K algorithm weighted by the distance of the nearest neighbors. (1/d) represents that the weight is obtained from the inverse of the distance ($1/distance$) whereas (1-d) means that the weight is obtained from the complemet of the distance ($1 - distance$). LWL is the implementation of the locally weighted learning algorithm proposed by Atkenson et al. [3]. Finally, K-Star is the implementation of the instance-based learner K* proposed by J. C. Cleary and L. E. Trigg [5].

Table 4 shows the classification rate (in %) comparison between WIB-K and other weighted and non weighted instance-based learners. WIB-K has achieved higher accuracy for all data sets except Dermatology and Hayes Roth. Even for the Dermatology and Hayes Roth data sets, WIB-K is better than many classifiers. Thus, in 6 out of 8 data sets, WIB-K has achieved the best performance. Table 4 also shows that the proposed algorithm WIB-K is consistently better than the original algorithms IB1 and IB-K.

Fig. 2 shows the classification rate with a bar graph. From the bar graph, it is clear that the proposed algorithm usually obtains superior results in terms of classification rate.

Table 4. Accuracy comparison between IB-k and other weighted and non weighted instance-based learners

DB	K	WIB-K	IB1	IB-K	dwIB-K (1/d)	dwIB-K (1-d)	LWL	K-Star
BS	24	**90.396**	79.027	89.436	89.436	89.436	53.932	88.474
BC	5	**97.216**	96.04	97.079	**97.216**	**97.216**	92.09	95.458
CMC	16	**55.126**	43.312	48.404	48.264	48.4	48.47	49.553
D	1	90.238	**95.269**	**95.269**	**95.269**	**95.269**	82.642	94.126
H	27	**75.483**	65.709	74.494	73.537	74.494	72.505	71.204
HR	2	75.604	76.538	62.087	67.417	67.417	**79.395**	61.263
LC	1	**70**	48.333	48.333	48.333	48.333	56.666	56.666
TAE	1	**72.958**	63.583	62.291	62.291	62.291	52.916	64.291

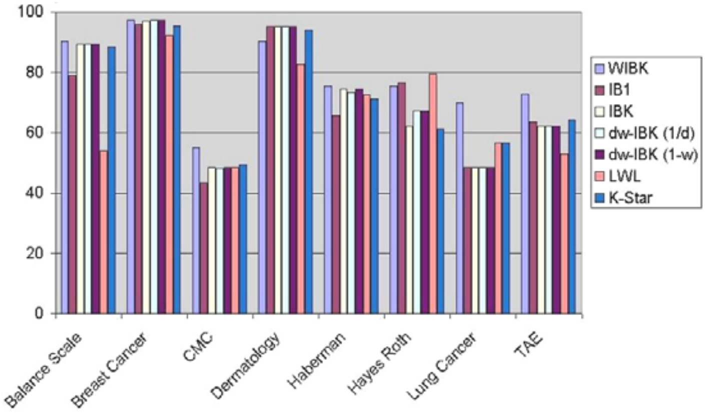

Fig. 2. Bar Graph of the results of Table 4

4.3 Comparison Against Well-known Classifiers

This subsection shows the results of the comparison of WIB-K against well-known and representative machine learning algorithms. NB is a naive bayes classifier that uses estimator classes [11]. SMO is the implementation of the algorithm to train a support vector classifier proposed by J. Platt [15]. MP is the implementation of a neural network that uses backpropagation to train. J48 is the implementation of the C4.5 decision tree proposed by R. Quinlan [16]. Finally, PART is a decision rule-based algorithm proposed by E. Frank and I. H. Witten [9].

Table 5. Accuracy comparison between IB-k and other well-known algorithms

DB Name	K	WIB-K	NB	SMO	MP	J48	PART
Balance Scale	24	90.39	90.04	87.68	**90.72**	76.64	83.52
Breast Cancer	5	**97.21**	96.33	97.07	96.04	96.04	95.46
CMC	16	**55.12**	49.28	50.98	54.51	53.22	50.10
Dermatology	1	90.23	97.48	**97.76**	97.48	95.25	93.29
Haberman	27	**75.48**	74.83	73.52	72.87	71.89	72.54
Hayes Roth	2	75.60	74.24	53.78	74.24	**80.30**	75.75
Lung Cancer	1	70	**70.37**	48.14	51.85	48.14	48.14
TAE	1	**72.95**	52.98	54.30	54.30	59.60	58.27

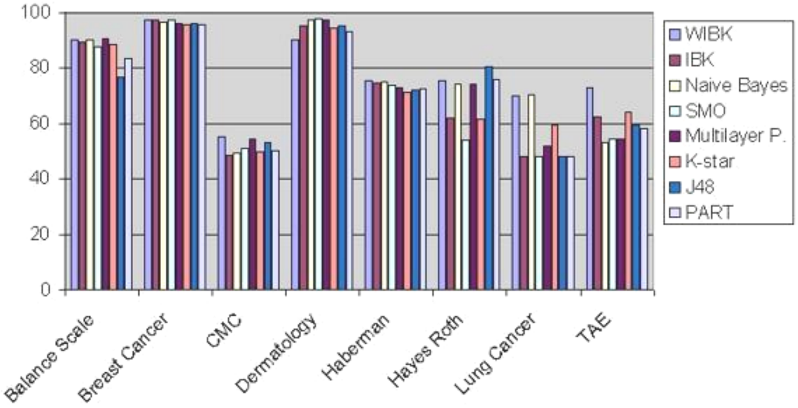

Fig. 3. Bar Graph for the results shown in Table 5

Table 5 shows the accuracy (in %) achieved by WIB-K and other representative machine learning algorithms. The WIB-K algorithm has achieved the highest accuracy in the Breast Cancer, CMC, Haberman, and TAE data sets, and, for the remaining data sets, WIB-K performed better than many well-known machine learning algorithms. The other algorithms obtained the best result in at most one data set whereas WIB-K did it in four data sets. In 4 out of 8 data sets WIB-K achieved the best performance.

Fig. 3 shows a bar graph for the data presented in Table 5. In the bar graph we can see that the WIB-K algorithm is highly competitive.

5 Conclusion and Future Work

In this paper we proposed a new weighted instance-based learning algorithm to perform classification of instances defined by integer valued features. This algorithm outputs one or more weights per feature for each class, and is noise tolerant. The weight is used in the distance function of the IB-K algorithm to improve accuracy rate.

The algorithm was tested on UCI databases of instances defined by integer attributes. The results indicate high competitiveness with respect to many well-known machine learning algorithms, as well as against weighted and non weighted instance-based learners.

The novelty of the algorithm relies in the approach to finds the intervals in which the value of a certain feature for a certain class falls, and obtains the weights directly from them. This approach opens new and interesting research paths. Knowing the interval in which the value of a certain feature for a certain class falls is important because it can give us more information about the data behavior.

Although this algorithm is restricted to integer-valued features, it is possible to apply it to real-valued features if, as a preprocessing step, the features are discretized [7]. Ordered nominal features can be directly converted into integers, however, the algorithm can not deal with non-ordered categorical features. Future work will focus on the search of a preprocessing scheme that allows WIB-K to deal with all kind of features.

Acknowledges The first author acknowledges CONACYT the support provided through the grant for Master's studies number 201804. The first author also acknowledges Erika Danaé López Espinoza for her valuable comments.

References

1. Aha, D.W.: Feature Weighting for Lazy Learning algorithms. Feature Extraction, Construction and Selection: A Data Mining Perspective, Vol. 1. The American Statistical Association, Boston, Massachusetts (1998)
2. Aha, D.W., Kibler, D., Albert, M.C.: Instance-Based Learning Algorithms. Machine Learning, Vol. 6. Springer-Verlag, Netherlands (1991) 37–66
3. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally Weighted Learning. Artificial Intelligence Review, Vol. 11. Springer-Verlag, Netherlands (1997) 11–73

4. Blansché, A., Gancarski, P., Korczak, J.J.: MACLAW: A modular approach for clustering with local attribute weighting. Pattern Recognition Letters, Vol. 27. Elsevier, Amsterdam (2006) 1299–1306
5. Cleary, J.G., Trigg, L.E.: K*: an instance-based learner using an entropic distance measure. Machine Learning: Proceedings of the Twelfth International Conference. Morgan Kaufmann, San Francisco (1995) 108–114
6. Cover, T.M., Hart, P.E.: Nearest Neighbor pattern classifier. IEEE Transactions on Information Theory, Vol. 13. IEEE Transactions on Information Theory Society, Los Alamitos California (1967) 21–27
7. Dougherty, J., Kohavi, Ron., Sahami, M.: Supervised and Unsupervised Discretization of Continuous Features. Machine Learning: Proceedings of the Twelfth International Conference. Morgan Kaufmann, San Francisco (1995)
8. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques. 2nd Edition. Morgan Kaufmann, San Francisco (2005)
9. Witten, I.H., Frank, E.: Generating Accurate Rule Sets Without Global Optimization. Machine Learning: Proceedings of the Fifteenth International Conference. Morgan Kaufmann, San Francisco (1998)
10. Gartner, T., Flach, P.A.: WBCSVM: Weighted Bayesian Classification based on Support Vector Machines. Machine Learning: Proceedings of the Eighteenth International Conference. Morgan Kaufmann, San Francisco (2001)
11. George, H., Langley, P.: Estimating Continuous Distributions in Bayesian Classifiers. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, Vol. 11. McGill University, Montreal, Quebec (1995) 338–345
12. Kira, K., Rendell, L.: The feature selection problem: traditional methods and new algorithm. Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI'92) The Association for the Advancement of Artificial Intelligence, San Jose, California (1992)
13. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. European Conference on Machine Learning (ECML-94). Springer-Verlag, Netherlands (1994)
14. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. University of California, California (1998)
15. Plat, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge, Massachusetts (1998)
16. Quinlan, J.R.: Induction of Decision Trees. Machine Learning, Vol. 1. Springer-Verlag, Netherlands (1986) 81–106
17. Smith, E.E., Medin, D.L.: Categories and Concepts. Hardvard University Press, Cambridge, Massachusetts (1981)
18. Tahir, T.A., Bouridane, A., Kurugollu, F.: Simultaneous feature selection and feature weighting using Hybrid Tabu Search/K-nearest neighbor classifier Pattern Recognition Letters, Vol. 28. Elsevier, Amsterdam (2007) 438–446
19. De la Torre, A., Peinado, A.M., Rubio, J.A., Segura, J.C., Benítez, C.: Discriminative feature weighting for HMM-based continuous speech recognizers Speech Communication, Vol. 38. Elsevier, Amsterdam (2002) 267–286
20. Weisstein, E.W.: The ANOVA test. Mathworld–A Wolfram web resource (2002) http://mathworld.wolfram.com/ANOVA.html
21. Wettschereck, D., Aha, D.W., Mohri, T.: A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms. Artificial Inteligence Reviews, Vol. 5. Springer-Verlag, Netherlands (1997) 273–314