ACKNOWLEDGMENT

The authors are grateful to reviewers for their insightful comments and suggestions.

REFERENCES

[1] J. Y. Hung, W. B. Gao, and J. C. Hung, "Variable structure control: A survey," *IEEE Trans. Ind. Electron.*, vol. 40, no. 1, pp. 2–22, 1993.
[2] S. C. Hwang and C. C. Kung, "A linguistic fuzzy-sliding mode controller," in *Proc. American Control Conf.*, Chicago, IL, 1992, pp. 1904–1905.
[3] S. B. Choi and J. S. Kim, "A fuzzy-sliding mode controller for robust tracking of robotic manipulators," *Mechatronics*, vol. 7, no. 2, pp. 199–216, 1997.
[4] G. C. Hwang and S. C. Lin, "A stability approach to fuzzy control design for nonlinear systems," *Fuzzy Sets Syst.*, vol. 48, pp. 279–287, 1992.
[5] Y. S. Lu and J. S. Chen, "A self-organizing fuzzy sliding-mode controller design for a class of nonlinear servo systems," *IEEE Trans. Ind. Electron.*, vol. 41, no. 5, pp. 492–502, 1994.
[6] F. Y. Hsu and L. C. Fu, "Adaptive robust fuzzy control for robot manipulators," in *Proc. IEEE Conf. Robotics Automation*, San Diego, CA, vol. 1, pp. 649–654, 1994.
[7] B. Yoo and W. Ham, "Adaptive fuzzy sliding mode control of nonlinear system," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 315–321, May 1998.
[8] M. R. Emami, A. A. Goldenberg, and I. B. Türksen, "A robust model-based fuzzy logic controller for robot manipulators," in *Proc. IEEE Conf. Robotics Automation*, Leuven, Belgium, 1998, pp. 2500–2505.
[9] ——, "Fuzzy logic dynamics modeling of robot manipulators," in *Proc. IEEE Conf. Robotics Automation*, Leuven, Belgium, 1998, pp. 2512–2517.
[10] S. C. Lin and Y. Y. Chen, "Design of self-learning fuzzy sliding mode controllers based on genetic algorithms," *Fuzzy Sets Syst.*, vol. 86, pp. 139–153, 1997.
[11] S. W. Kim and J. J. Lee, "Design of a fuzzy controller with fuzzy sliding surface," *Fuzzy Sets Syst.*, vol. 71, pp. 359–367, 1995.
[12] Z. M. Yeh, "A performance approach to fuzzy control design for nonlinear systems," *Fuzzy Sets Syst.*, vol. 64, pp. 339–352, 1994.
[13] L. X. Wang, "Design and analysis of fuzzy identifiers of nonlinear dynamic systems," *IEEE Trans. Automat. Contr.*, vol. 40, pp. 11–23, Jan. 1995.
[14] R. Palm, "Sliding mode fuzzy control," in *Proc. IEEE Conf. Fuzzy Systems*, San Diego, CA, 1992, pp. 519–526.
[15] R. M. Sanner and J. J. E. Slotine, "Gaussian networks for direct adaptive control," *IEEE Trans. Neural Networks*, vol. 3, pp. 837–863, Nov. 1992.
[16] F. C. Sun, Z. Q. Sun, and P. Y. Woo, "Stable neural network-based adaptive control for sampled-data nonlinear systems," *IEEE Trans. Neural Networks*, vol. 9, pp. 956–968, Sept. 1998.

Considerations About Sample-Size Sensitivity of a Family of Edited Nearest-Neighbor Rules

Francesc J. Ferri, Jesús V. Albert, and Enrique Vidal

***Abstract*— The edited nearest neighbor classification rules constitute a valid alternative to *k*-NN rules and other nonparametric classifiers. Experimental results with synthetic and real data from various domains and from different researchers and practitioners suggest that some editing algorithms (especially, the optimal ones) are very sensitive to the total number of prototypes considered. This paper investigates the possibility of modifying optimal editing to cope with a broader range of practical situations. Most previously introduced editing algorithms are presented in a unified form and their different properties (and not just their asymptotic behavior) are intuitively analyzed. The results show the relative limits in the applicability of different editing algorithms.**

***Index Terms*—Edited NN rule, nearest neighbors (NN), nonparametric classification, prototype selection,**

I. INTRODUCTION

The k-nearest neighbor (k-NN) classification rule consists of finding the k nearest neighbors to each target point according to a certain dissimilarity measure (not necessarily a distance) and making a decision according to the (known) classification of these k neighbors, usually by assigning the label of the most voted class among these neighbors. A trivial case of this rule is when $k = 1$, in which each point is assigned to the same class as its NN.

Regardless of the measure used, the asymptotic classification error of the k-NN rule (when the number of prototypes, n, tends to infinity) tends to the Bayes error rate, R^*, as $k \to \infty$ and $k/n \to 0$. If $k = 1$, the error is bounded by approximately $2R^*$ [1]. This behavior in classification performance combines with the simple and comparatively inexpensive training (in terms of computational burden) which only requires gathering correctly classified prototypes, to make a powerful classification technique capable of dealing with arbitrarily complex problems, provided there is a large enough set of prototypes available.

In spite of this quality in performance and training requirements, the NN rules also exhibit some practical disadvantages. An obvious drawback comes from the impossibility of having a sufficiently large number of prototypes to achieve (near) optimal results in practice because there is no way of fulfilling the requirements for k and n at the same time. On the other hand, in the (fortunate) case of having large sets of prototypes, it necessarily implies a significant computational burden to find the (k)-nearest neighbor(s) and makes the NN methods inapplicable for problems in which dissimilarity calculation is a time-consuming procedure in itself. In order to overcome these drawbacks, a number of different approaches have already been proposed, e.g., fast searching algorithms [2], weighted NN rules [3], optimal distance measures [4], and prototype selection techniques [1].

Manuscript received April 14, 1999. This work was supported in part by Projects TIC98-677-C02-02 and 1FD97-279. This paper was recommended by Associate Editor K. Pattipati.

F. J. Ferri and J. V. Albert are with the Department Informàtica i Electrònica, Universitat de València, 46100 Burjassot València, Spain.

E. Vidal is with the Department Sistemi Informàtics i Computació, Universitat Politècnica de València, 46100 Burjassot València, Spain.

Publisher Item Identifier S 1083-4419(99)07950-9.

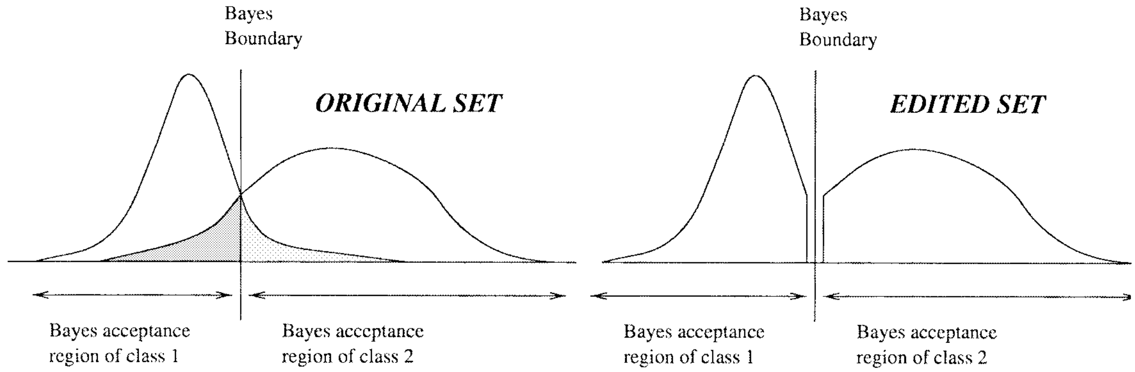

Fig. 1. Illustration of the optimal editing procedure.

Prototype selection techniques were first introduced by Hart [5] more than two decades ago when he introduced the *Condensing* algorithm. This algorithm obtains a *reduced* and *consistent* (that correctly classifies the original set) set of prototypes to be used with the 1-NN rule without significantly degrading the corresponding performance when using the original set. The Condensed NN rule improves the efficiency of the resulting classifier but only constitutes an approximation to the plain NN rule in terms of classification performance. The problem of selecting a subset of prototypes from a given input set in such a way that this subset leads to an *improved* performance when used with the plain 1-NN rule is usually referred to as *Editing* [1], [6].

The first work on Editing was that of Wilson [7], and many others followed [8]–[10] leading to the work of Devijver and Kittler who introduced the well-known Multiedit algorithm. The purpose of Editing is two-fold: first, to remove possible outliers which strongly degrade the performance of the NN rules, and second, to approximate the Bayes-optimal performance (asymptotically exhibited by the k-NN rule) by means of the simple 1-NN rule. Another good reason for applying these techniques is the way it combines with the condensing procedures. In the presence of outliers or with strong overlap among classes, condensing often produces sets containing an arbitrarily large number of "bad" prototypes, i.e., performance degrades and not much reduction is achieved. Conversely, if the set of prototypes has been previously edited, the number of condensed prototypes tends to be drastically reduced and the final (expected) performance is usually close to that of the edited rule and superior to the one with the plain 1-NN rule. Thus, Editing constitutes a compulsory step prior to applying any condensing procedure and, usually, both steps are considered as a unified technique [1].

The general editing idea along with a generalized scheme is presented in the following section. In further subsections, the classical Editing algorithms are presented and the small-sample case is taken into account. Also, appropriate modifications of the asymptotically optimal editing procedures are considered from a practical point of view. Section III includes experiments with both synthetic and real data which were designed to study and compare the behavior of different algorithms. The final section presents the main conclusions drawn from the results obtained from the experiments and also includes some comments about possible extensions to the work.

II. Edited Nearest Neighbor Rules

The idea of Editing relies on the fact that one can optimally eliminate outliers and possible overlap among classes from a given set of prototypes, so that the training of the corresponding classifier becomes easier in practice. According to Fig. 1, where the apparent probability density functions of two classes (after and before editing) are shown, the prototypes falling in the Bayes acceptance regions of a different class need to be removed in order to obtain well clustered clouds of prototypes that tightly define a decision boundary as close to the optimal one as possible.

In the context of distance-based classifiers, this implies that, with a sufficient number of prototypes, the straightforward 1-NN rule can become as powerful as the optimal Bayes classifier if applied after such an editing procedure [1].

One inherent problem comes from the fact that the Bayes acceptance regions can only be approximately obtained by using an estimate of the true class label of each prototype in the original set. Usually the k-NN rule or the Parzen estimate have been used in this context for this purpose. Also, from a practical point of view, it is not possible to remove only prototypes lying in wrong acceptance regions without also removing some "correct" prototypes. In practice, this leads to suboptimal results. Most editing algorithms constitute different kinds of tradeoffs between removing too many (even correct) prototypes and leaving some small overlap among classes. In the latter case, Editing can also be applied in a repetitive way. These approaches assume that the first (possibly suboptimal) edited set can be further improved by iterating the very same procedure. In some specific cases, this fact can be formally proved under certain assumptions [1].

A. The Generalized Editing Scheme

The above approach to the prototype selection problem leads to a generalized scheme that is valid for any classification rule, estimate, and stopping criterion [11]

Generalized Editing Scheme

Let $\mathcal{R}$ be the initial set of prototypes. δ, ξ, and σ are the classification rule, error estimator, and stopping criterion, respectively

1. Using the (error-counting) estimator ξ, obtain an estimate of the classification error for the rule δ trained with the set $\mathcal{R}$. Let $\mathcal{S}$ be the set of prototypes misclassified in this process.
2. Let $\mathcal{R} = \mathcal{R} - \mathcal{S}$
3. if σ then STOP, else go to 1.

Traditionally, different error-counting estimates have been used for editing purposes, mainly, an adaptation of the Holdout estimate and different realizations of the Crossvalidation estimate including Leaving-one-out. These estimates are only used in this context to decide which prototypes need to be removed. The error-rate itself is not used at all. As a consequence, other improved error-counting estimates based on resampling as bootstrap and jackknife [12] are not directly applicable according to the above generalized scheme and will not be considered in this paper.

B. Editing Procedures Based on Internal Estimates

The first estimate used for editing purposes was the Leaving-one-out [7]. In this work, a simple nonrepetitive editing algorithm was proposed along with a theoretical analysis of the behavior of the edited NN rule. This Editing procedure can be summarized as follows:

Wilson Editing Algorithm

Let $\mathcal{R}$ be the initial set of prototypes

1. Let $\mathcal{S}$ be the subset of prototypes, p, that are misclassified using the k-NN rule with $\mathcal{R} - \{p\}$.
2. Return $\mathcal{R} - \mathcal{S}$.

This technique brought about a number of different algorithms which consisted of slight changes of the classification rule used [9], [10]. Some of the results of the analysis carried out by Wilson were disproved by Penrod and Wagner [8] who pointed out the difficulties of obtaining an exact analysis of the Edited NN Rules. In their work, Devijver and Kittler threw new light on this controversy and the facts that make the analysis difficult were clearly identified and an (optimal) way of circumventing the underlying problems was proposed [1].

Taking into account that the prototypes in the initial set are alternatively used as "test" (p) and "train" ($\mathcal{R} - \{p\}$) prototypes, it follows that *statistical independence* (as postulated in Wilson's analysis) cannot be assumed and this was the reason for obtaining optimistically biased bounds [8]. To achieve this statistical independence, the classification of prototypes can be performed in a *Holdout* manner. This means that, ideally, "test" and "train" prototypes have to be obtained independently and, moreover, their functions cannot be interchanged. It can be proved that the corresponding Holdout editing is asymptotically optimal in the Bayes sense [1].

Holdout editing could be implemented by randomly partitioning the initial set of prototypes but, in this case, only "test" prototypes could be edited. Instead, the concepts of *diffusion* (splitting the set into several, more than 2, independent random samples) and *confusion* (pooling the different results into a new set) are used to effectively remove the undesirable statistical dependence and to eliminate prototypes from each block using only two independent blocks at the same time.

As using the k-NN rule with this "modified" Holdout was not of practical use (due to the strong dependency on the parameter k), a repetitive version, the Multiedit algorithm, was also proposed

Multiedit Algorithm

Let $\mathcal{R}$ be the initial set of prototypes

1. Let $\mathcal{S} = \emptyset$ and randomly split $\mathcal{R}$ into B blocks , $\mathcal{R}_1, \cdots, \mathcal{R}_B$, where[1] $B > 2$
2. *For* $b = 1, \cdots, B$ *do*
   Add to $\mathcal{S}$ the prototypes from $\mathcal{R}_b$ that are misclassified using the k-NN rule with $\mathcal{R}_{(b+1)\bmod B}$
3. If $\mathcal{S} = \mathcal{R}$ during the last I iterations, STOP
4. Let $\mathcal{R} = \mathcal{S}$, go to 1.

[1]The setting $B = 2$ is not allowed because it constitutes a particular case of cross validation in which test prototypes and train prototypes are interchanged.

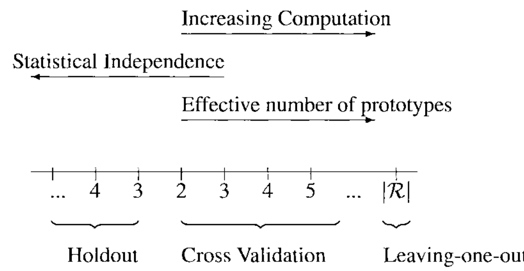

Fig. 2. Behavior of Holdout and Cross Validation estimates (as used in the editing algorithms) with respect to the number of blocks B, in which the set $\mathcal{R}$ is partitioned.

In successive iterations, the Multiedited NN rule converges to the optimal classifier when applied to infinite sets of prototypes. Nevertheless, as noted by various researchers, in practice the editing algorithms based on the Holdout estimate behave poorly when used with finite sets of prototypes [11], [13]–[15].

C. Editing Finite Sets of Prototypes

Another family of algorithms based on the B-fold Crossvalidation (CV) estimate [12] was proposed as a way of improving the behavior of the Holdout-based editing in the finite sample size case [11]. The corresponding algorithm consists of a slight modification which implies a larger effective number of prototypes used for estimation purposes. Algorithmically, it can be expressed as follows:

CV Multiedit Algorithm

Let $\mathcal{R}$ be the initial set of prototypes

1. *Let* $\mathcal{S} = \emptyset$ and randomly split $\mathcal{R}$ into B blocks, $\mathcal{R}_1, \cdots, \mathcal{R}_B$
2. *For* $b = 1, \cdots, B$ *do*
   Add to $\mathcal{S}$ the prototypes from $\mathcal{R}_b$ that are misclassified using the k-NN rule with $\bigcup_{j \neq b} \mathcal{R}_j$
3. If $\mathcal{S} = \mathcal{R}$ during the last I iterations, STOP
4. Let $\mathcal{R} = \mathcal{S}$, go to 1.

Each iteration of this algorithm can be considered as a compromise between Holdout and Wilson's algorithm. In fact, the three estimates involved can be considered as belonging to the same family in which statistical independence and effective number of prototypes change according to the values of the parameter B. From this point of view, Cross Validation can be seen as a different realization of the ideal Holdout that preserves the effective number of prototypes rather than the statistical independence.

The behavior of the estimates used in editing algorithms can be summarized as in Fig. 2. According to our experience, Cross Validation appears to be a good tradeoff between a randomized asymptotically optimal estimate (Holdout) and a deterministic estimate that behaves well in practice (Leaving-one-out).

A summary of most of the editing algorithms already proposed along with their corresponding settings for δ and ε in the generalized scheme is shown in Table I. The only stopping criterion considered corresponds to "I iterations without change." The small illustrations show how the initial set, $\mathcal{R}$, is internally partitioned to estimate the true classification of prototypes. The grey part of the small figures

TABLE I
SETTINGS FOR DIFFERENT EDITING ALGORITHMS ACCORDING TO THE GENERALIZED SCHEME. THE COMPUTATIONAL COST PER ITERATION IS EXPRESSED IN TERMS OF $n = |\mathcal{R}|$

Rule (δ)	**Error Estimator** (ξ)	**Iterate**	**Algorithm**	**Optimal**	**Cost**
k-NN	Leaving-one-Out test train	NO	Wilson Editing [7]	-	$O(n^2)$
		YES	Unlimited Editing [9]	-	
ℓ-NN $\ell=1,..,k$		NO	*All-k* Editing [9]	-	
(k, k')-NN		NO	kk' Editing [10]	-	
		NO	Editing with Reject[1]	-	
k-NN	Holdout test train	NO	Holdout Editing [1]	$k\to\infty$	$O(n^2/B)$
1-NN		YES	Multiedit [1]	$I\to\infty$	
k-NN	Cross Validation test train	NO	CV Editing	-	$O(\frac{B-1}{B}n^2)$
1-NN		YES	CV Multiedit	-	

corresponds to the effective number of prototypes used as training. The computational complexity of each of the methods depends on this size as well as on the number of iterations for repeated editing algorithms (see Fig. 2).

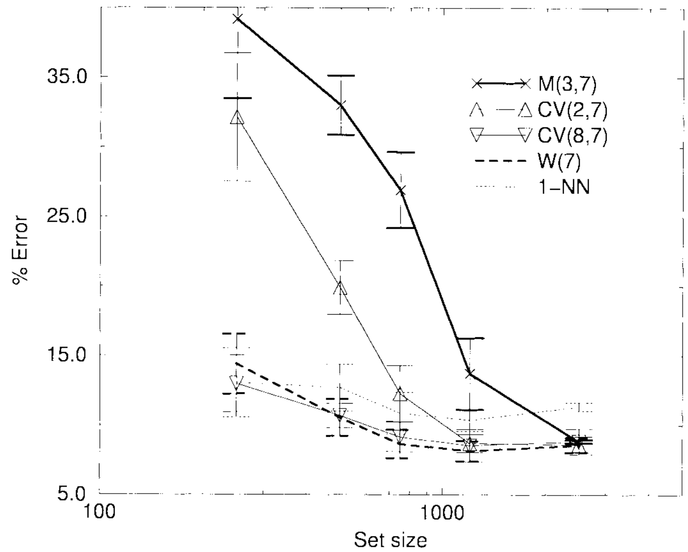

Fig. 3. Error rate of different editing algorithms for a two-spiral problem with Gaussian noise.

III. EXAMPLES AND EXPERIMENTS

Several experiments with both synthetic and real data were carried out in order to study the behavior of some of the different editing algorithms discussed above. In particular, *no editing* (1-NN), *Wilson Editing* (W(k)), *Repeated Holdout Editing* (or Multiedit, M(B, I)) and *Repeated Crossvalidation Editing* (CV(B, I)), were considered. In these methods, k is the number of neighbors for the k-NN rule, B is the number of blocks in the partition and I is the required number of iterations without change. The goal of these experiments was to show the sensitivity to the number of prototypes for the different editing algorithms, taking into account the intrinsic dimensionality or problem complexity.

Each one of the following two synthetic-data experiments was carried out as follows: a random sample of fixed size was generated and the error rate for each editing technique was estimated for five different random partitions of the sample into training and test sets. This step was repeated several times (depending on the sample size) with different random samples. The results reported correspond to the average over all these repeated trials. Only one typical set of parameters ($B = 3$; $k = 7$; $I = 7$) was considered for each editing algorithm apart from the proposed modifications ($B = 2$, $B = 8$). For the real-world data experiments considered, only one sample set (presumably too small with respect to the underlying statistics) was available. In this case, the editing algorithms were tested with several different settings for the values of their parameters.

In order to obtain more reliable results, for editing algorithms involving internal randomization, the results were computed by taking the average over five trials with different seeds for the pseudorandom generators used.

Synthetic data: First, a two-class problem in two dimension with complex decision boundaries was considered. The problem consisted of two embedded spirals given by $x_1(t) = 10t\sin(t)$, $y_1(t) = 10t\cos(t)$ for class 1, and $x_2(t) = -x_1(t)$, $y_2(t) = -x_1(t)$ for class 2, $t \in [0, 10]$.

From this distribution, samples of 250, 500, 750, 1250, and 2500 points were drawn with additive bivariate Gaussian noise, given by $\sigma_{11}(t) = \sigma_{22}(t) = t+3.5$. Experiments with these sets were repeated 30, 15, ten, six, and three times, respectively. The results are shown in Fig. 3. For all editings based on the Holdout estimator, including our two-fold CV-editing (which is equivalent to a "degenerate" Multiedit with $B = 2$) a rapid increase in error-rate is observed as the sample size gets smaller. This behavior is not exhibited by Wilson Editing or by the B-fold CV-Editing, with $B = 8$, which seems to be the best option (although the difference is not statistically significant).

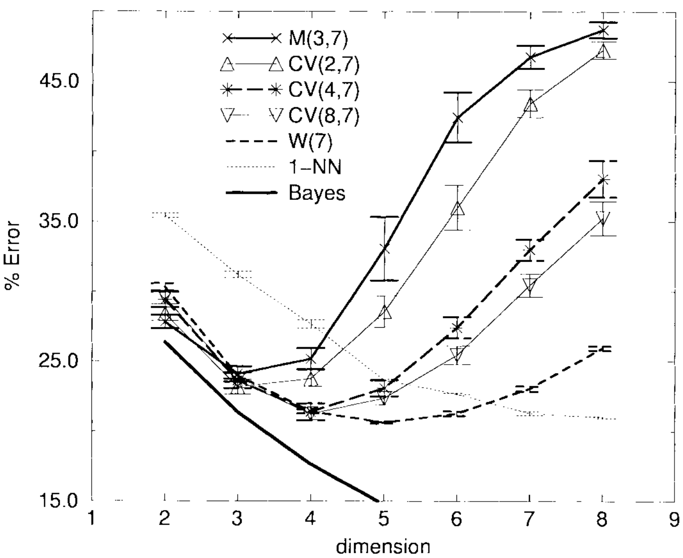

Fig. 4. Error rate of different editing algorithms with respect to the 1-NN rule and the Bayes classifier for a two-class Gaussian classification problem. [Wilson (W), Holdout (H), Multiedit (M), and CV-Multiedit (CV)].

Second, a set of seven two-class problems with dimensionality ranging from two to eight and a fixed number of prototypes (2500 per class) was considered. The problem consisted of two multivariate normal distributions with zero mean. The standard deviations in all dimensions were one and two, respectively. For this particular problem, the Bayes error can be computed exactly and is shown along with the editing results.

The overlap between classes makes this problem difficult for the editing algorithms. In fact, most of the prototypes of one of the classes need to be discarded in order to obtain an optimal edited set. A behavior close to the previous experiment can be seen in Fig. 4. In this case, all the repetitive algorithms gave bad results for high dimensionalities while the Wilson's algorithm got worse more slowly. There is not enough statistical evidence to say

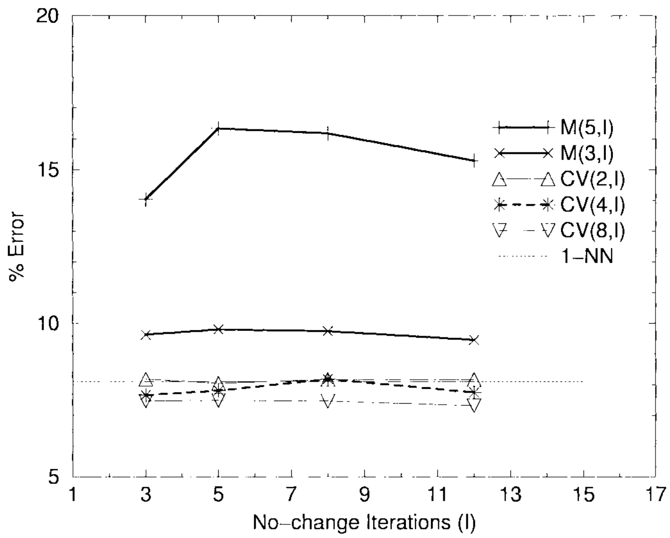

Fig. 5. Error rate of different editings for a multispeaker isolated word recognition experiment consisting of ten different words.

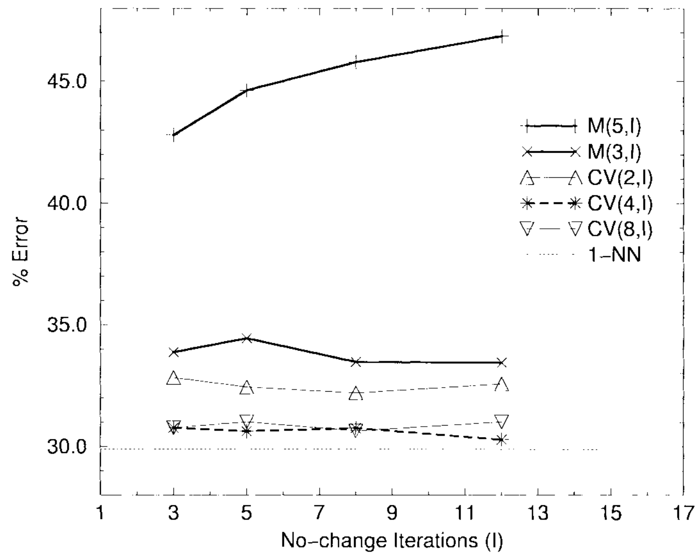

Fig. 6. Error rate of different editings for a multispeaker isolated word recognition experiment using the Spanish E-set.

which algorithm was better at low dimensionalities and none of the algorithms considered were better than the plain 1-NN rule at high dimensionalities. Nevertheless, what is absolutely clear from the figure is the general tendency of the algorithms based on different estimates to degenerate as the ratio of the sample size to intrinsic dimensionality decreases. For example, in dimension 4, the results from the methods based on Holdout and CV with different values of B clearly follow a behavior that can be easily explained from the behavior of the corresponding estimates shown in Fig. 2. It is also worth noting that the worst algorithm from this point of view, the Multiedit algorithm, is (marginally) the best at dimension 2.

Real Data: A set of 1000 isolated word utterances corresponding to ten different Spanish letters was considered. The utterances were spoken by ten different male and female speakers and consisted of the words (given by their phonetic transcription) /ace/, /i/, /xota/, /ka/, /ele/, /eme/, /ene/, /o/, /pe/, /ku/. In addition, another set of 900 isolated word utterances from the (most difficult) Spanish E-set vocabulary, which was spoken by the same ten speakers, was considered. This set consists of the words /efe/, /ele/, /eλe/, /eme/, /ene/, /eηe/, /ere/, /e$\underline{\text{r}}$e/, and /ese/. Using standard parameterization techniques similar to those used in Castro *et al.* [16], these utterances were converted into variable-length strings of 11-dimensional vectors of cepstral coefficients. The metric adopted to compare such representations was given by a conventional symmetric nonslope-constrained Dynamic Time Warping procedure [17].

The results corresponding to the first set of utterances is shown in Fig. 5. It can be observed that Repetitive Editing algorithms clearly follow the behavior observed in previous synthetic experiments. Multiedit obtains much worse results than its CV-based counterpart. Moreover, the CV results are better for higher values of B as expected from Fig. 2. The results using Wilson's editing exhibited a variable behavior depending on k, but were never better than the best results shown in Fig. 5.

The results corresponding to the second, more difficult set, are shown in Fig. 6, which confirms the tendency exhibited in the previous figure. In particular, Multiedit dramatically degrades with B, and again the CV-Multiedit constitutes a "better" choice when compared to the Multiedit algorithm. It is interesting to note that, in this case, *none* of the Repetitive algorithms led to better results than the 1-NN rule. For this problem, the result with Wilson's editing (not shown here) exhibited a decrease in performance as k increased. The Wilson editing with $k = 3$ gave the only result which was better than the one obtained with the 1-NN rule. This suggests that the set which was available for this particular experiment (900 points) is not large enough for properly applying the currently available editing techniques. The results obtained in both real data experiments reproduce the same situation as with the previously shown synthetic experiments.

If we compare the results obtained in the last two experiments with those represented in Fig. 3, it is possible to observe a similarity between the results obtained for small set sizes in Fig. 3 and the ones obtained in the two real data experiments. The results obtained in the second real-data experiment and the fact that it is possible to obtain significantly better results [16] using different approaches (involving different parameterization and preprocessing techniques) with *the same number of prototypes* suggest that the edited NN rule in such a situation could be further improved.

IV. Conclusions and Further Extensions

In this paper, Editing algorithms derived from the original work of Wilson (and consequently based on internal estimation of the misclassification rate) have been presented in a unified way. Optimal properties of the editing algorithms have been considered and their behavior under the small sample size assumption has been studied and illustrated with both synthetic and real experiments.

There is enough empirical evidence to conclude that different editing algorithms have differences in sensitivity to the number of prototypes used or, in other words, to the ratio of the sample size to intrinsic dimensionality. It is very important to be aware of this behavior to properly apply the different algorithms in critical situations.

In summary, our results clearly indicate that improved editing techniques are required for the cases in which only small samples are available which, in practice, are unfortunately too often the case. One of the key facts is that the failure of the algorithms in the small sample case mainly stems from the inability of different error-rate estimators to achieve sufficiently reliable estimates with the data available. In general, one can strongly relax (or even completely sacrifice) the *statistical independence* assumption (as in CV or Wilson's editings), so as to take the *maximum advantage of the data* available in order to

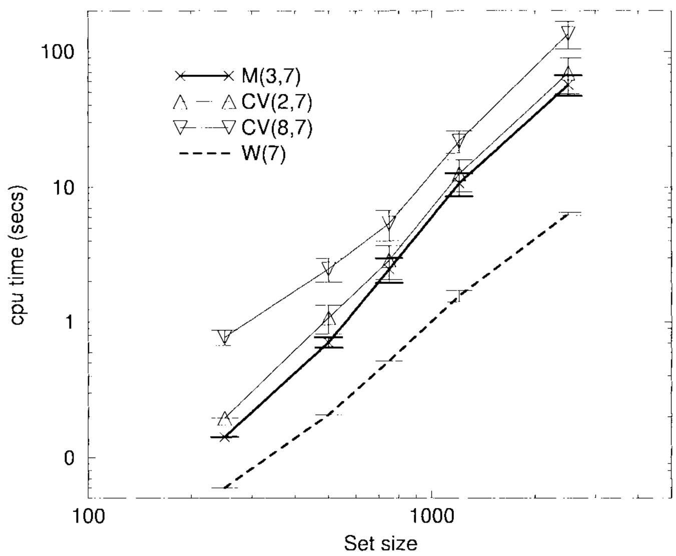

Fig. 7. CPU times in seconds (on a R5000-based workstation at 180 MHz) corresponding to the different editing algorithms used when applied to the two-spiral problem.

boost the reliability of the estimators; this generally gives improved and more robust results.

From a practical standpoint, the computational burden that improved estimates imply is worth the benefits in performance, especially taking into account that editing is usually applied off-line. In Fig. 7, it can be seen that the quadratic behavior of the different algorithms in the first synthetic experiment is modified only by a constant which directly depends on the number of iterations (which in turn depends on the complexity of the problem [1]) rather than on the estimator used. It is worth pointing out that even though the Leaving-one-out editing gives very good results in critical situations with minimum computation loads, the potential benefits of obtaining random edited sets instead of deterministic ones make the CV editings interesting enough in practice. This must be considered together with the fact that both Holdout and CV-based editing clearly improve the Leaving-one-out version for large data sets.

Recent developments on editing attempt to improve the internal estimate by applying the basic idea in a random way using genetic algorithms [15], or by looking into alternatives for the k-NN rule [13]. These ideas could be combined using alternative, specially-adapted, error estimates to better tune the trade-off between the error-estimation reliability versus statistical-independence.

References

[1] P. A. Devijver and J. Kittler, *Pattern Recognition. A Statistical Approach.* Englewood Cliffs, NJ: Prentice-Hall, 1982.

[2] B. V. Dasarathy and B. V. Sheela, "Visiting nearest neighbors," in *Proc. Int. Conf. Cybern. Soc.*, 1977, pp. 630–635.

[3] J. Macleod, A. Luck, and D. Titterington, "A re-examination of the distance-weighted k-nearest-neighbor classification rule," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-17, pp. 689–696, July/Aug. 1987.

[4] R. Short and K. Fukunaga, "The optimal distance measure for nearest neighbor classification," *IEEE Trans. Inform. Theory,* vol. IT-27, no. 5, pp. 622–627, 1981.

[5] P. E. Hart, "The condensed nearest neighbor rule," *IEEE Trans. Inform. Theory,* vol. IT-14, pp. 515–516, 1968.

[6] Devroye, Gyorfi, and Lugosi, *A Probabilistic Theory of Pattern Recognition.* Berlin, Germany: Springer-Verlag, 1995.

[7] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-2, pp. 408–421, May/June 1972.

[8] C. Penrod and T. Wagner, "Another look at the edited nearest neighbor rule," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, pp. 92–94, 1977.

[9] I. Tomek, "An experiment with the edited nearest neighbor rule," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 2, pp. 121–126, 1976.

[10] J. Koplowitz and T. Brown, "On the relation of performance to editing in nearest neighbor rules," *Pattern Recognit.*, vol. 13, no. 3, pp. 251–255, 1981.

[11] F. Ferri and E. Vidal, "Small sample size effects in the use of editing techniques," in *Proc. 11th Int. Conf. Pattern Recognition,* The Hague, The Netherlands, pp. 607–610, Sept. 1992.

[12] S. Raudys and A. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 252–264, MAr. 1991.

[13] J. S. Sanchez, F. Pla, and F. J. Ferri, "Prototype selection for the nearest neighbor rule through proximity graphs," *Pattern Recognit. Lett.*, vol. 18, no. 6, pp. 507–513, 1995.

[14] N. Yalabik, F. Yarman-Vural, and A. Mansur, "Modified condensed nearest neighbor rule as applied to speaker independent word recognition," *Speech Commun.*, vol. 7, pp. 411–415, 1988.

[15] L. Kuncheva, "Editing for the k-nearest neighbors rule by a genetic algorithm," *Pattern Recognit. Lett.*, vol. 16, no. 8, pp. 809–814, 1995.

[16] M. J. Castro and F. Casacuberta, "The use of multilayer perceptrons in isolated word recognition," in *Artificial Neural Networks. Lecture Notes in Computer Science,* vol. 540, pp. 348–354, 1991.

[17] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASAP-26, pp. 43–49, 1978.

Comments on "Constraints on Belief Functions Imposed by Fuzzy Random Variables": Some Technical Remarks on Römer/Kandel

C. Römer and A. Kandel

First, we would like to thank V. Kratschmer for his validation of our results in the above paper[1] regarding the belief measure by using a topological approach. Though Assertions (1) and (3) are presented in a weakened fashion, our results still remain valid, as he claims. It is true that Assertion (2) has been proved by us only for Borel sets B, which have at most countable components. We were not able to prove the same result for Borel sets with uncountable components (such as the irrational numbers, for example) using our line of reasoning. We therefore applaud the proof presented by V. Kratschmer for the more general Borel sets using an interesting use of some topological properties induced by the Hansdorff metric defined on the space of closed intervals of the real numbers. This certainly makes our original approach to fuzzy data analysis combining fuzzy sets theory and Dempster–Shafer even more useful.

Manuscript received April 1, 1998. This paper was recommended by Associate Editor A. Sage.

The authors are with the Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620-5399.

Publisher Item Identifier S 1083-4419(99)00908-5.

[1] V. Krätschmer, Trans. Syst., Man, Cybern. B, vol. 28, pp. 881–883, Dec. 1998.