Using Representative-Based Clustering for Nearest Neighbor Dataset Editing

Christoph F. Eick
Dept. of Computer Science
University of Houston
ceick@cs.uh.edu

Nidal Zeidat
Dept. of Computer Science
University of Houston
nzeidat@cs.uh.edu

Ricardo Vilalta
Dept. of Computer Science
University of Houston
vilalta@cs.uh.edu

Abstract

The goal of dataset editing in instance-based learning is to remove objects from a training set in order to increase the accuracy of a classifier. For example, Wilson editing removes training examples that are misclassified by a nearest neighbor classifier so as to smooth the shape of the resulting decision boundaries. This paper revolves around the use of representative-based clustering algorithms for nearest neighbor dataset editing. We term this approach supervised clustering editing. The main idea is to replace a dataset by a set of cluster prototypes. A novel clustering approach called supervised clustering is introduced for this purpose. Our empirical evaluation using eight UCI datasets shows that both Wilson and supervised clustering editing improve accuracy on more than 50% of the datasets tested. However, supervised clustering editing achieves four times higher compression rates than Wilson editing; moreover, it obtains significantly high accuracies for three of the eight datasets tested.

Keywords: nearest neighbor editing, instance-based learning, supervised clustering, representative-based clustering, clustering for classification, Wilson editing.

1. Introduction

Nearest Neighbor classification (also called 1-NN-Rule) was first introduced by Fix and Hodges in 1951 [4]. Given a set of *n* classified examples in a dataset *O*, a new example *q* is classified by assigning the class of the nearest example $x \in O$ using some distance function *d*.

$$d(q,x) \leq d(q,o_i)\ o_i \in O \qquad (1)$$

Since its birth, the 1-NN-Rule and its generalizations have received considerable attention by the research community. Most research aims at producing time-efficient versions of the algorithm (for a survey see Toussaint [8]). Many partial distance techniques and efficient data structures have been proposed to speed up nearest neighbor queries. Furthermore, several condensing techniques have been proposed that replace the set of training examples *O* by a smaller set $O_C \subset O$ such that all examples in *O* are still classified correctly by a NN-classifier that uses O_C.

Replacing a dataset *O* with a usually smaller dataset O_E with the goal of improving the accuracy of a NN-classifier belongs to a set of techniques called *dataset editing*. The most popular technique in this category is called *Wilson editing* [10] (see Fig. 1); it removes all examples that have been misclassified by the 1-NN rule from a dataset. Wilson editing cleans interclass overlap regions, thereby leading to smoother boundaries between classes. Figure 2.a shows a hypothetical dataset where examples that are misclassified using the 1-NN-rule are marked with circles around them. Figure 2.b shows the reduced dataset after applying Wilson editing.

```
PREPROCESSING
A. For each example $o_i \in O$
   1. Find the $k$-nearest neighbors of $o_i$ in $O$ (excluding
      $o_i$)
   2. Classify $o_i$ with the class associated with the largest
      number of examples among the $k$-nearest neighbors
      (breaking ties randomly)
B. Edit dataset $O$ be deleting all examples that were
   misclassified in step A.2.

CLASSIFICATION RULE
Classify a new example $q$ using k-NN classifier using the
   edited subset $O_E$ of $O$.
```

Figure 1: Wilson's Dataset Editing Algorithm.

It has been shown by Penrod and Wagner [7] that the accuracy of a Wilson edited nearest neighbor classifier converges to Bayes error as *n* approaches infinity. But even though Wilson editing was proposed more than 30 years ago, the benefits of such technique s regards to data mining have not been explored systematically by past research.

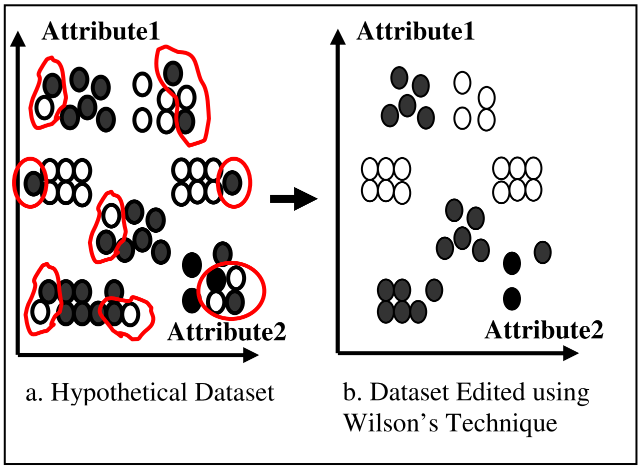


Figure 2: Wilson Editing for a 1-NN Classifier.

Devijver and Kittler [2] proposed an editing technique they call *multi-edit* that repeatedly applies Wilson editing to random partitions of the data set until a predefined termination condition is met. Moreover, several variations of Wilson editing have been proposed for k-nearest neighbor classifiers (e.g. in Hattori and Takahashi [5]). Finally, the relationship between condensing and editing techniques has been systematically analyzed in the literature (see for example Dasaranthy, Sanchez, and Townsend [1]).

In addition to analyzing the benefits of Wilson editing, this paper proposes a new approach based on using representative-based clustering algorithms for nearest neighbor editing. The idea is to replace a dataset by a set of cluster prototypes. A new data set editing technique is proposed that applies a *supervised clustering* algorithm [11] to the dataset, and uses the resulting cluster representatives as the output of the editing process. We will refer to this editing technique as *supervised clustering editing* (SCE); we will refer to the corresponding nearest neighbor classifier as *nearest representative (NR) classifier*. Unlike traditional clustering, supervised clustering is applied on classified examples with the objective of identifying clusters that maximize the degree of class purity within each cluster. Supervised clustering seeks to identify regions on the attribute space that are dominated by instances of a single class, as depicted in Fig. 3.b.

The remainder of this paper is organized as follows. Section 2 introduces supervised clustering and explains how supervised clustering dataset editing works. Section 3 discusses experimental results that compare Wilson editing, supervised clustering editing, and traditional, ”unedited” nearest-neighbor classifiers, with respect to classification accuracy and dataset reduction rates. Section 4 summarizes the results of this paper and identifies areas of future research.

A summary of the notations used throughout the paper is given in Table 1.

Notation	Description
$O=\{o_1, \ldots, o_n\}$	Objects in a dataset (training set)
n	Number of objects in the dataset
$d(o_i,o_j)$	Distance between objects o_i & o_j
c	The number of classes in the dataset
C_i	Cluster associated with the i-th representative
$X=\{C_1, \ldots, C_k\}$	A clustering solution consisting of clusters C_1 to C_k
$k=\|X\|$	The number of clusters (or representatives) in a clustering solution X
$q(X)$	A fitness function that evaluates a clustering X, see formula (2)

Table 1: Notations Used in the Paper.

2. Using Supervised Clustering for Dataset Editing

Due to its novelty, the goals and objectives of supervised clustering will be discussed in the first subsection. The second subsection introduces representative-based supervised clustering algorithms. Finally, we will explain how supervised clustering can be used for nearest neighbor dataset editing.

2.1 Supervised Clustering

Clustering is typically applied in an unsupervised learning framework using particular error functions, e.g. an error function that minimizes the distances inside a cluster. Supervised clustering, on the other hand, deviates from traditional clustering in that it is applied on classified examples with the objective of identifying clusters having not only strong cohesion but also class purity. Moreover, in supervised clustering, we try to keep the number of clusters small, and objects are assigned to clusters using a notion of closeness with respect to a given distance function.

The fitness functions used for supervised clustering are quite different from the ones used by *traditional* clustering algorithms. Supervised clustering evaluates a clustering based on the following two criteria:

- *Class impurity, Impurity(X).* Measured by the percentage of minority examples in the different clusters of a clustering X. A minority example is an example that belongs to a class different from the most frequent class in its cluster.
- *Number of clusters, k.* In general, we favor a low number of clusters; but clusters that only contain a single example are not desirable, although they maximize class purity.

In particular, we use the following fitness function in our experimental work (lower values for q(X) indicate 'better' quality of clustering X)**.**

$$q(X) = \text{Impurity}(X) + \beta * \text{Penalty}(k) \qquad (2)$$

where

$$\text{Impurity}(X) = \frac{\text{\# of Minority Examples}}{n},$$

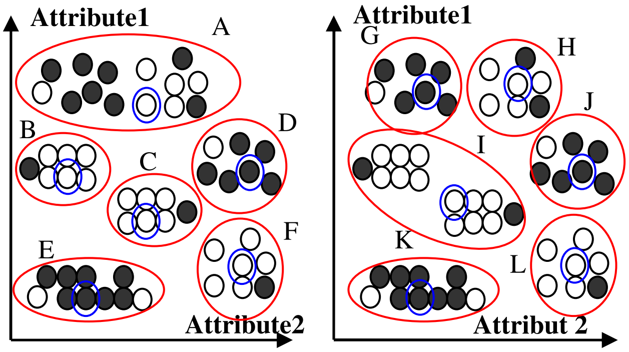
$$\text{Penalty}(k) = \begin{cases} \sqrt{\dfrac{k-c}{n}} & k \geq c \\ 0 & k < c \end{cases}$$

with *n* being the total number of examples and *c* being the number of classes in a dataset. Parameter β (0< β ≤3.0) determines the penalty that is associated with the numbers of clusters, *k*: higher values for β imply larger penalties as the number of clusters increases.

Two special cases of the above fitness function should be mentioned; the first case is a clustering X1 that uses only *c* clusters; the second case is a clustering X2 that uses *n* clusters and assigns a single object to each cluster, therefore making each cluster pure. We observe that q(X1)=Impurity(X1) and q(X2)≈β.

Finding the best, or even a good, clustering X with respect to the fitness function *q* is a challenging task for a supervised clustering algorithm due to the following reasons (these matters have been discussed in more detail in [3,11]):

1. The search space is very large, even for small datasets.
2. The fitness landscape of *q* contains a large number of local minima.
3. There are a significant number of ties[1] in the fitness landscape creating plateau-like structures that present a major challenge for most search algorithms, especially hill climbing and greedy algorithms.



a. Dataset clustered using a traditional clustering algorithm

b. Dataset clustered using a supervised clustering algorithm.

Figure 3: Traditional and Supervised Clustering.

Fig. 3 illustrates the differences between traditional and supervised clustering. Let us assume that the black examples and the white examples in the figure represent subspecies of Iris plants named Setosa and Virginica, respectively. A traditional clustering algorithm, such as the k-medoid algorithm [6], would, very likely, identify the six clusters depicted in Figure 3.a. Cluster representatives are encircled. If our objective is to generate summaries for the Virginica and Setosa classes of the Iris Plant, for example, the clustering in Figure 3.a would not be very attractive since it combines Setosa (black circles) and Virginica objects (white circles) in cluster **A** and allocates examples of the Virginica class (white circles) in two different clusters **B** and **C,** although these two clusters are located next to each other.

A supervised clustering algorithm that maximizes class purity, on the other hand, would split cluster **A** into two clusters **G** and **H**. Another characteristic of supervised clustering is that it tries to keep the number of clusters low. Consequently, clusters **B** and **C** would be merged into one cluster without compromising class purity while reducing the number of clusters. A supervised clustering algorithm would identify cluster **I** as the union of clusters **B** and **C** as depicted in Figure 3.b.

2.2 Representative-Based Supervised Clustering Algorithms

Representative-based clustering aims at finding a set of *k* representatives that best characterize a dataset. Clusters are created by assigning each object to the closest representative. Representative-based supervised clustering

[1] Clusterings X1 and X2 with q(X1)=q(X2).

algorithms seek to accomplish the following goal: *Find a subset O_R of O such that the clustering X obtained by using the objects in O_R as representatives minimizes q(X).*

One might ask why our work centers on developing *representative-based* supervised clustering algorithms. The reason is representatives (such as medoids) are quite useful for data summarization. Moreover, clustering algorithms that restrict representatives to objects belonging to the dataset, such as the *k-medoid* algorithm, Kaufman [6], explore a smaller solution space if compared with centroid–based clustering algorithms, such as the *k-means* algorithm[2]. Finally, when using representative-based clustering algorithms, only an inter-object distance matrix is needed and no "new" distances have to be computed, as it is the case with *k-means*.

As part our research, we have designed and evaluated several supervised clustering algorithms [3]. Among the algorithms investigated, one named Single Representative Insertion/Deletion Steepest Decent Hill Climbing with Randomized Restart (SRIDHCR for short) performed quite well[3]. This greedy algorithm starts by randomly selecting a number of examples from the dataset as the initial set of representatives. Clusters are then created by assigning examples to their closest representative. Starting from this randomly generated set of representatives, the algorithm tries to improve the quality of the clustering by adding a single non-representative example to the set of representatives as well as by removing a single representative from the set of representatives. The algorithm terminates if the solution quality (measured by *q*(X)) does not show any improvement. Moreover, we assume that the algorithm is run *r* (input parameter) times starting from a randomly generated initial set of representatives each time, reporting the best of the *r* solutions as its final result. The pseudo-code of the version of SRIDHCR that was used for the evaluation of supervised clustering editing is given in Figure 4. It should be noted that the number of clusters *k* is not fixed for SRIDHCR; the algorithm searches for "good" values of *k*.

```
REPEAT r TIMES
    curr := randomly generated set of
    representatives with size between c+1 and 2*c
    WHILE NOT DONE DO
        1. Create new solutions S by adding a
           single non-representative to curr and
           by removing a single representative
           from curr
        2. Determine the element s in S for
           which q(s) is minimal (if there is
           more than one minimal element,
           randomly pick one)
        3. IF q(s)<q(curr) THEN curr:=s
           ELSE IF q(s)=q(curr) AND |s|>|curr|
           THEN curr:=s
           ELSE terminate and return curr as the
           solution for this run.
Report the best out of the r solutions found.
```

Figure 4: Pseudo Code of SRIDHCR.

2.3 Using Cluster Prototypes for Dataset Editing

In this paper we propose using supervised clustering as a tool for *editing* a dataset *O* to produce a reduced subset O_r. The subset O_r consists of cluster representatives that have been selected by a supervised clustering algorithm. A 1-NN classifier, that we call nearest-representative (NR) classifier, is then used for classifying new examples using subset O_r instead of the original dataset *O*. Figure 5 presents the classification algorithm that the NR classifier employs. A NR classifier can be viewed as a compressed 1-nearest-neighbor classifier because it uses only *k* (*k*<*n*) examples out of the *n* examples in the dataset *O*.

```
PREPROCESSING
A. Apply a representative-based supervised
   clustering algorithm (e.g. SRIDHCR) on dataset
   O to produce a set of k prototypical examples.
B. Edit dataset O by deleting all non-representative
   examples to produce subset O_r.

CLASSIFICATION RULE
Classify a new example q by using a 1-NN classifier
with the edited subset O_r.
```

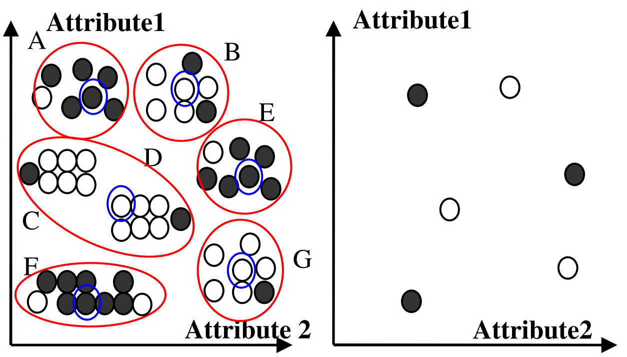
Figure 5: Nearest Representative (NR) Classifier.

Figure 6 gives an example that illustrates how supervised clustering is used for dataset editing. Figure 6.a shows a dataset that was partitioned into 6 clusters using a supervised clustering algorithm. Cluster representatives

[2] There are 2^n possible centroids for a dataset containing n objects.

[3] Another algorithm named SCEC [12] that employs evolutionary computing to evolve a population consisting of sets of representatives, also denoted good performance.

are marked with circles around them. Figure 6.b shows the result of supervised clustering editing.



a. Dataset clustered using supervised clustering.

b. Dataset edited using cluster representatives.

Figure 6: Editing a Dataset Using Supervised Clustering.

3. Experimental Results

To evaluate the benefits of Wilson editing and supervised clustering editing (SCE), we applied these techniques to a benchmark consisting of 8 datasets that were obtained from the UCI Machine Learning Repository [9]. Table 2 gives a summary of these datasets.

All datasets were normalized using a linear interpolation function that assigns 1 to the maximum value and 0 to the minimum value. Manhattan distance was used to compute the distance between two objects.

Dataset name	# of objects	# of attributes	# of classes
Glass	214	9	6
Heart-Statlog	270	13	2
Heart-Disease-Hungarian (Heart-H)	294	13	2
Iris Plants	150	4	3
Pima Indians Diabetes	768	8	2
Image Segmentation	2100	19	7
Vehicle Silhouettes	846	18	4
Waveform	5000	21	3

Table 2: Datasets Used in the Experiments.

Parameter β has a strong influence on the number k of representatives chosen by the supervised clustering algorithm; i.e., the size of the edited dataset O_r. If high β values are used, clusterings with a small number of representatives are likely to be chosen. On the other hand, low values for β are likely to produce clusterings with a large number of representatives.

In general, an editing technique reduces the size n of a dataset to a smaller size k. We define the dataset compression rate of an editing technique as:

$$\text{Compression Rate } = 1 - \frac{k}{n} \qquad (3)$$

In order to explore different compression rates for supervised clustering editing, three different values for parameter β were used in the experiments: 1.0, 0.4, and 0.1.

Prediction accuracies were measured using 10-fold cross-validation throughout the experiments for the four classifiers tested. Representatives for the nearest representative (NR) classifier were computed using a version of the SRIDHCR supervised clustering algorithm that was introduced in Section 2.2. In our experiments, SRIDHCR was restarted 50 times (r = 50), each time with a different initial set of representatives, and the best solution (i.e., set of representatives) found in the 50 runs was used as the edited dataset for the NR classifier. Accuracies and compression rates were obtained for a 1-NN-classifier that operates on subsets of the 8 datasets obtained using Wilson editing. We also computed prediction accuracy for a traditional 1-NN classifier that uses all training examples when classifying a new example. The reported accuracies of the traditional 1-NN-classifier serve as a baseline for evaluating the benefits of the two editing techniques. Finally, we also report prediction accuracy for decision-tree learning algorithm C4.5 that was run using its default parameter settings. Table 3 reports the accuracies obtained by the four classifiers evaluated in our experiments.

Table 4 reports the average dataset compression rates for supervised clustering editing and Wilson editing. Due to the fact that the supervised clustering algorithm has to be run 10 times, once for each fold, different numbers of representatives are usually obtained for each fold. Consequently, Table 4, also, reports the average, minimum, and maximum number of representatives found on the 10 runs. For example, when running the NR classifier for the Diabetes dataset with β set to 0.1 the (rounded) average number of representatives was 27, the maximum number of representatives during the 10 runs was 33 and the minimum number of representatives was 22; supervised clustering editing reduced the size of the original dataset O by an average of 96.5%, as displayed in Table 4. The NR classifier classified 73.6% of the testing examples correctly, as indicated in Table 3. Table 4 only reports average compression rates for Wilson editing. Minimum and maximum compression rates observed in different folds are not reported, because the deviations among these numbers were quite small.

β	NR	Wilson	1-NN	C4.5
Glass* (214*)				
0.1	0.636	0.607	0.692	0.677
0.4	0.589	0.607	0.692	0.677
1.0	0.575	0.607	0.692	0.677
***Heart-Stat Log* (270)**				
0.1	0.796	0.804	0.767	0.782
0.4	0.833	0.804	0.767	0.782
1.0	0.838	0.804	0.767	0.782
***Diabetes* (768)**				
0.1	0.736	0.734	0.690	0.745
0.4	0.736	0.734	0.690	0.745
1.0	0.745	0.734	0.690	0.745
***Vehicle* (846)**				
0.1	0.667	0.716	0.700	0.723
0.4	0.667	0.716	0.700	0.723
1.0	0.665	0.716	0.700	0.723
***Heart-H* (294)**				
0.1	0.755	0.809	78.33	80.22
0.4	0.793	0.809	78.33	80.22
1.0	0.809	0.809	78.33	80.22
***Waveform* (5000)**				
0.1	0.834	0.796	0.768	0.781
0.4	0.841	0.796	0.768	0.781
1.0	0.837	0.796	0.768	0.781
***Iris-Plants* (150)**				
0.1	0.947	0.936	0.947	0.947
0.4	0.973	0.936	0.947	0.947
1.0	0.953	0.936	0.947	0.947
Segmentation (2100)				
0.1	93.81	0.966	0.956	0.968
0.4	91.9	0.966	0.956	0.968
1.0	88.95	0.966	0.956	0.968

Table 3: Predition Accuracy for the four Algorithms.

If we inspect the results displayed in Table 3, we can see that Wilson editing is a quite useful technique for improving traditional 1-NN-classfiers. Using Wilson editing leads to higher accuracies for 6 of the 8 datasets tested (e.g., Heart-StatLog, Diabetes, Vehicle, Heart-H, Waveform, and Segmentation) and only shows a significant loss in accuracy for the Glass dataset. The SCE approach, on the other hand, accomplished significant improvement in accuracy for the Heart-Stat Log, Waveform, and Iris-Plants datasets, outperforming Wilson editing by at least 2% in accuracy for those datasets. It should also be mentioned that the achieved accuracies are significantly higher than those obtained by C4.5 for those datasets. However, our results also indicate that SCE does not work well for all datasets. A significant loss in accuracy can be observed for the Glass and Segmentation datasets.

β	Avg. *k* [Min-Max] for SCE	SCE Compression Rate	Wilson Compression Rate
Glass* (214*)			
0.1	34 [28-39]	84.3	27
0.4	25 [19-29]	88.4	27
1.0	6 [6 – 6]	97.2	27
***Heart-Stat Log* (270)**			
0.1	15 [12-18]	94.4	22.4
0.4	2 [2 – 2]	99.3	22.4
1.0	2 [2 – 2]	99.3	22.4
***Diabetes* (768)**			
0.1	27 [22-33]	96.5	30.0
0.4	9 [2-18]	98.8	30.0
1.0	2 [2 – 2]	99.74	30.0
***Vehicle* (846)**			
0.1	57 [51-65]	97.3	30.5
0.4	38 [26-61]	95.5	30.5
1.0	14 [9-22]	98.3	30.5
***Heart-H* (294)**			
0.1	14 [11-18]	95.2	21.9
0.4	2	99.3	21.9
1.0	2	99.3	21.9
***Waveform* (5000)**			
0.1	104 [79-117]	97.9	23.4
0.4	28 [20-39]	99.4	23.4
1.0	4 [3-6]	99.9	23.4
***Iris-Plants* (150)**			
0.1	4 [3-8]	97.3	6.0
0.4	3 [3 – 3]	98.0	6.0
1.0	3 [3 – 3]	98.0	6.0
Segmentation (2100)			
0.1	57 [48-65]	97.3	2.8
0.4	30 [24-37]	98.6	2.8
1.0	14	99.3	2.8

Table 4: Dataset Compression Rates for SCE and Wilson Editing .

More importantly, looking at Table 4, we notice that with the exception of the Glass and the Segmentation datasets, SCE accomplishes compression rates of more than 95% without a significant loss in prediction accuracy for the other 6 datasets. For example, for the Waveform dataset, a 1-NN classifier that only uses 28 representatives outperforms the traditional 1-NN classifier that uses all 4500 training examples[4] by 7.3% points in accuracy, increasing the accuracy from 76.8% to 84.1%. Similarly, for the Heart-StatLog dataset, a 1-NN classifier that uses just one representative for each class outperforms C4.5 by

[4] Due to the fact that we use 10-fold cross-validation training sets contain 0.9*5000=4500 examples.

more than 5% points, and the traditional 1-NN classifier by more than 6% points.

As mentioned earlier, Wilson editing reduces the size of a dataset by removing examples that have been misclassified by a *k*-NN classifier. Consequently, the data set reduction rates are quite low on "easy" classification tasks for which high prediction accuracies are normally achieved. For example, Wilson editing produces dataset reduction rates of only 2.8% and 6.0% for the Segmentation and Iris datasets, respectively. Most condensing approaches, on the other hand, reduce the size of a dataset by removing examples that have been classified correctly by a nearest neighbor classifier. Finally, supervised clustering editing reduces the size of a dataset by removing examples that have been classified correctly as well as examples that have not been classified correctly. A representative-based supervised clustering algorithm is used that aims at finding clusters that are dominated by instances of a single class, and tends to pick as the cluster representative[5] objects that are in the center of the region associated with the cluster. As depicted in Fig. 6, supervised clustering editing just keeps the cluster representative and removes all other objects belonging to a cluster from the dataset. Furthermore, it seeks to minimize the fitness function $q(X)$ rather than considering which objects have been or have not been classified correctly by a *k*-nearest neighbor classifier.

It can also be seen that the average compression rate for Wilson editing is approximately 20%, and that supervised clustering editing obtained compression rates that are usually at least four times as high. Prior to conducting the experiments we expected that the NR classifier would perform better for lower compression rates. However, as can be seen in Table 4, this is not the case: for six of the eight datasets, the highest accuracies were obtained using β=0.1 or β=0.4, and only for two datasets the highest accuracy was obtained using β=1.0. For example, for the Diabetes dataset using just 2 representatives leads to the highest accuracy of 74.5%, whereas a 1-NN classifier that uses all 768 objects in the dataset achieves a lower accuracy of 69%. The accuracy gains obtained using a very small number of representatives for several datasets are quite surprising.

We also claim that our approach of associating a generic penalty function with the number of clusters has clear advantages when compared to running a clustering algorithm keeping the number of clusters, *k*, fixed. Parameter β narrows the search space to values of *k* corresponding to "good" solutions, but does not restrict it to a single value. Consequently, a supervised clustering algorithm still tries to find the best value of *k* within the boundaries induced by β without the need for any prior knowledge of what values for *k* are "good" on a particular dataset.

4. Conclusion

The goal of dataset editing in instance-based learning is to remove objects from a training set in order to increase the accuracy of the learnt classifier. In contrast to condensing techniques, editing techniques have not received much attention in the machine learning and data mining literature. One popular dataset editing technique is Wilson editing. It removes those examples from a training set that are misclassified by a nearest neighbor classifier. In this paper, we evaluate the benefits of Wilson editing using a benchmark consisting of eight UCI datasets. Our results show that Wilson editing enhanced the accuracy of a traditional nearest neighbor classifier on six of the eight datasets tested. Wilson editing achieved an average compression rate of about 20%. It is also important to note that Wilson editing, although initially proposed for nearest neighbor classification, can easily be used for other classification tasks. For example, a dataset can easily be "*Wilson edited*" by removing all training examples that have been misclassified by a decision tree classification algorithm.

In this paper, we introduced a new technique for dataset editing called supervised clustering editing (SCE). The idea of this approach is to replace a dataset by a subset of cluster prototypes. We introduced a novel clustering approach, called supervised clustering, that determines clusters and cluster prototypes in the context of dataset editing. Supervised clustering, itself, aims at identifying class-uniform clusters that have high probability densities.

Using supervised clustering editing, we implemented a 1NN-classifier, called nearest representative (NR) classifier. Experiments were conducted that compare the accuracy and compression rates of the proposed NR classifier, with a 1-NN classifier that employs Wilson editing, and with a traditional, unedited, 1-NN classifier. Results show that the NR-classifier accomplished significant improvements in prediction accuracy for 3 out of the 8 datasets used in the experiments, outperforming the Wilson editing based 1-NN classifier by more than 2%. Moreover, experimental results show that for 6 out the 8 datasets tested, SCE achieves compression rates of more than 95% without significant loss in accuracy. We also explored using very high compression rates and its

[5] Representatives are rarely picked at the boundaries of a region dominated by a single class, because boundary points have the tendency to attract points of neighboring regions that are dominated by other classes, therefore increasing cluster impurity.

effect on accuracy. We observed that high accuracy gains were achieved using only a very small number of representatives for several datasets. For example, for the Waveform dataset, a traditional 1-NN classifier that uses all 5000 examples accomplished an accuracy of 76.8%. The NR-classifier, on the other hand, uses only an average of 28 examples, and achieved an accuracy of 84.1%. In summary, our empirical results stress the importance of centering more research on dataset editing techniques.

Our future work will focus on 1) using data set editing with other classification techniques, 2) making data set editing techniques more efficient, and 3) exploring the relationships between condensing techniques and supervised clustering editing. We also plan to make our supervised clustering algorithms readily available on the web.

References

[1] Dasarathy, B.V., Sanchez, J.S., and Townsend, S., "*Nearest neighbor editing and condensing tools – synergy exploitation*", Pattern Analysis and Applications, 3:19-30, 2000.

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

[3] Eick, C., Zeidat, N., and Zhao, Z., "*Supervised Clustering - Objectives and Algorithms.* submitted for publication.

[4] Fix, E. and Hodges, J., "*Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties*", Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas, 1951.

[5] Hattori, K. and Takahashi, M., "A new edited k-nearest neighbor rule in the pattern classification problem", Pattern Recognition, 33:521-528, 2000.

[6] Kaufman, L. and Rousseeuw, P. J., "*Finding Groups in Data: an Introduction to Cluster Analysis*", John Wiley & Sons, 1990.

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

[8] Toussaint, G., "*Proximity Graphs for Nearest Neighbor Decision Rules: Recent Progress*", Proceedings of the 34th Symposium on the INTERFACE, Montreal, Canada, April 17-20, 2002.

[9] University of California at Irving, Machine Learning Repository. http://www.ics.uci.edu/~mlearn/MLRepository.html

[10] Wilson, D.L., "*Asymptotic Properties of Nearest Neighbor Rules Using Edited Data*", IEEE Transactions on Systems, Man, and Cybernetics, 2:408-420, 1972.

[11] Zeidat, N., Eick, C., "*Using k-medoid Style Algorithms for Supervised Summary Generation*", Proceedings of MLMTA, Las Vegas, June 2004.

[12] Zhao, Z., "*Evolutionary Computing and Splitting Algorithms for Supervised Clustering*", Master's Thesis, Dept. of Computer Science, University of Houston, May 2004.