A data driven stopping criterion for evolutionary instance selection

Walter D. Bennette

Abstract Instance based classifiers, such as k-Nearest Neighbors, predict the class value of a new observation based on some distance or similarity measure between the new instance and the stored training data. However, due to the required distance calculations, classifying new instances becomes computationally expensive as the number of training observations increases. Therefore, instance selection techniques have been proposed to improve instance based classifiers by reducing the number of training instances that must be stored to achieve adequate classification rates. Although other methods exist, an evolutionary algorithm has been used for instance selection with some of the best results in regard to data reduction and preservation of classification accuracy. Unfortunately, the performance of the evolutionary algorithm for instance selection comes at the cost of longer computation times in comparison to classic instance selection techniques. In this work we introduce a new stopping criterion for the evolutionary algorithm which depends on the convergence of its fitness function. Experimentation shows that the new criterion results in less computation time while achieving comparable performance.

1 Introduction

Classification is a supervised machine learning task where labeled training data is used to predict the class value of previously unlabeled instances (data points) based on observed attribute values. Instance based classifiers, such as *k*-Nearest Neighbors (kNN), predict the class value of a new observation based on some distance or similarity measure between the new instance and the stored training data [1, 6].

Walter D. Bennette
The United States Air Force, 525 Brooks Road Rome NY 13441, e-mail: walter.bennette.1@us.af.mi

DISTRIBUTION A: Approved for public release: distribution unlimited: 02 May 2016. Case # 88ABW-2016-2258l

However, due to the required distance calculations, classifying new instances becomes computationally expensive as the number of training observations increases. Therefore, Instance Selection (IS) techniques have been proposed to improve instance based classifiers by reducing the number of training instances that must be stored to achieve adequate classification rates [14].

The objective of IS is to find a reduced set of instances to include in an instance-based classifier's training dataset without sacrificing predictive accuracy. Although other methods exist, evolutionary Algorithms (EA) have been used for IS with some of the best results in regard to data reduction and preservation of classification accuracy [14]. Unfortunately, the performance of EAs for IS comes at the cost of longer computation times in comparison to classic IS techniques [4]. Additionally, evolutionary IS exhibits prohibitive computational time and slow convergence when applied to large datasets [4, 5, 3, 7, 14]. Although approaches exist to scale IS to large datasets [3, 7, 8], we believe improving the de facto evolutionary IS algorithm can contribute to the mitigation of these issues.

The de facto evolutionary IS algorithm was established in a 2003 study by Cano et al [4]. In this study four representative EA models were compared and it was determined that the *Cross generational elitist selection, Heterogeneour recombination and Cataclysmic mutation* (CHC) [10] EA was the best for IS in regard to data reduction, classification accuracy, and computation time (amongst other EAs) [4]. This study is oft cited to motivate the use of CHC for IS. However, the stopping criterion used for CHC in these studies is a set number of generations, and as shown in Table 1, mostly the same as the original 2003 study. We argue that 1) the number of generations used by most studies is too large, as in practice the fitness function plateaus much earlier than the prescribed 10,000 generations, and that 2) a stopping criterion that looks at the convergence of the fitness function, as discussed in [21], can be used to reduce the computational burden of applying CHC to a dataset. Such an approach is justified as it is apparent from a search of the literature that no study has been conducted to find an appropriate number of generations for termination, and preliminary results indicate that the required number of generations is dependent on the dataset (this can also be seen from an inspection of Figure 1).

Reference	Date	Generations	Population
[4]	2003	10,000	50
[11]	2006	10,000	50
[5]	2006	10,000	50
[3]	2007	10,000	50
[7]	2009	100	100
[13]	2009	10,000	50
[8]	2012	1,000	100
[12]	2012	10,000	50
[16]	2013	Unknown	Unknown
[19]	2013	1,000 & 100	50
[14]	2015	10,000	50

Table 1 Parameters from studies that use CHC for IS.

The remainder of this paper is organized as follows: the Background section provides relevant information about IS and the formulation of IS for CHC; the New Stopping Criterion section motivates and introduces the new stopping criterion; in the Experimental Study section our new stopping criterion is applied to experimental datasets; and finally, the Conclusion summarizes the effort and provides avenues for future work.

2 Background

In the following subsections we will introduce IS and the CHC algorithm.

2.1 Instance Selection

The process of IS is utilized for instance based classifiers, such as kNN, because faster classifications can be made by maintaining only certain necessary instances in the classifier's training dataset [17, 20, 24]. That is, because instance based classifiers perform distance calculations for each instance of a dataset every time a new classification is to be made, a smaller dataset requires a fewer number of overall calculations. It has also been observed that IS not only reduces dataset size, but can improve dataset quality by removing outliers, noise, or instances with contradictory class values [18, 22, 26].

One of the earliest IS methods was Condensed Nearest Neighbors (CNN), introduced by Hart in 1968 [17]. CNN operates by iteratively building a subset of the training data such that all instances of the original training data are still correctly classified by the reduced set. Edited Nearest Neighbors [24] closely followed CNN as a more selective technique that removes instances that do not have a class value that agrees with their k-nearest neighbors. Both techniques tend to improve classification accuracy by removing border instances and making decision boundaries easier to find, but add relatively little value for data reduction.

Alternatively, DROP3 boasts higher data reduction rates while maintaining acceptable levels of classification accuracy [25]. This method requires a greater number of calculations and determines a selection by comparing the classification accuracy of specific neighborhoods with and without the presence of different training instances. Finally, the CHC EA typically achieves the best data reduction and classification accuracy, but does so with the greatest computational expense [14]. CHC will be discussed in greater detail in the following subsection.

One disadvantage of all IS methods is very long (or prohibitive) run times for large datasets [14]. Scaling approaches have been suggested to mitigate this issue, and the majority involve performing IS on several independent or nearly independent subsamples of the original data [3, 7, 8]. Such approaches scale IS by reducing the amount of data, and therefore the number of distance calculations, that the individual IS methods must consider. This reduces the total number of required operations.

In depth reviews of IS can be found in [14] and [15].

2.2 CHC Evolutionary Algorithm

As defined in [4], given a training set, TR, with m instances, IS can be represented by an evolutionary algorithm such that an individual chromosome represents a subset, S, of TR. Each of a chromosome's m binomial genes corresponds to an instance from TR and holds the value one if it is to be included in S and zero if it is not. The fitness function is then a combination of the classification accuracy and the percent reduction of instances between S and TR,

$$Fitness(S) = \alpha * clasAccuracy + (1-\alpha) * percReduction.$$

In this work we use the 3-NN classifier to calculate the classification accuracy, clasAccuracy, of S. Specifically we let *clasAccuracy* represent the percentage of instances correctly labeled in TR when using S to find the nearest neighbor. Percent reduction, *perReduction*, is calculated as,

$$perReduction = 100 * \frac{|TR| - |S|}{|TR|}.$$

In the Experimental Study section we use $\alpha = 0.5$, as is standard across the literature.

First described in [4] for IS, each generation of CHC utilizes the following steps.

1. A parent population of N individuals is randomly paired to produce a child population of N individuals. N is generally arbitrarily chosen to be 50 or 100.
2. A survival competition is held and the N individuals from the parent and offspring populations with the highest fitness function value are selected to create the next generation.

Members of the child population are generated using heterogeneous recombination, HUX. HUX operates by exchanging half of the bits that are different between the parent chromosomes, where the bits chosen to be swapped are randomly selected. Also, the hamming distance between two chromosomes is calculated before applying HUX, and only parents that differ by a certain threshold are crossed. This serves as a form of incest prevention. To begin, the crossing threshold is set at $L/4$, where L is the length of the chromosomes. The crossing threshold is reduced by one if no child is introduced into the population at the end of step two.

Note that mutation is not applied during the generation of children. Instead, when the population begins to stagnate, as noted by the crossing threshold dropping to zero, a cataclysmic re-population occurs to inject diversity into the search. A cataclysmic re-population is realized by removing all chromosomes from the population except the chromosome with the highest fitness value. The best chromosome is then used to generate $N-1$ new members of the population by randomly changing 35% of its bits. The search continues after a cataclysmic re-population.

The stopping criterion used by Cano et al. for the CHC algorithm is to simply terminate the procedure after 10,000 generations. Although heavily adopted, this criterion is seemingly arbitrarily chosen. We will show in the next section that this stopping criterion can result in more generations than required to achieve an acceptable solution.

3 New Stopping Criterion

Rather than terminating CHC after 10,000 generations, we propose a criterion that looks at the convergence of the CHC fitness function. Specifically, after each cataclysmic re-population we calculate the difference in fitness of the best chromosome from the last cataclysmic re-population and the fitness of the best chromosome from the current cataclysmic re-population. That is, we measure how much the fitness function has increased during the generations between cataclysmic re-populations. If this change is less than or equal to some threshold, G, CHC is terminated. This way we are using feedback from the dataset and the search itself to terminate; not an arbitrarily chosen value.

To help determine an appropriate value of G, CHC is applied to the five tuning datasets described in Table 2. For each dataset we record where CHC would terminate for $G \in \{1.0, 0.75, 0.5, 0.25, 0.0\}$, the result of which can be seen in Figure 1 and Figure 2. Figure 1 shows that regardless of the chosen G, 10,000 generations far exceeds where the fitness function plateaus for the small datasets. Figure 2 shows the opposite in that 10,000 generations is far before the fitness function plateaus for the medium, “Pageblocks” dataset.

Dataset	Attributes	Instances	Classes
Mamographic	5	961	2
Monk-2	6	432	2
Pageblocks	10	5404	10
Spectfheart	44	267	2
Wisconsin	9	699	2

Table 2 Tuning datasets for determining the threshold, G, of the proposed stopping criterion.

It has been noted that CHC could have difficulty converging for large datasets [4, 5, 3, 7, 14], and Figure 2 supports this claim as it takes over 60,000 generations for CHC to terminate when $G = 0.0$. To alleviate convergence and computational difficulties from large datasets, most authors recommend a scaling approach. Therefore we performed the stratified approach from [4] and recorded the evolution of each fold. Briefly stated the stratified approach from [4] splits the data into inde-

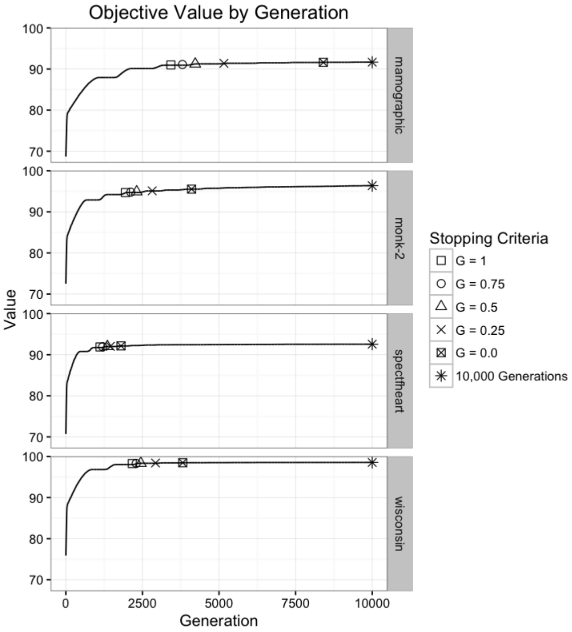

Fig. 1 Fitness of CHC for test datasets by generation.

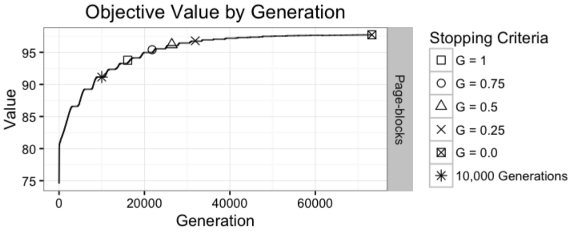

Fig. 2 Fitness of CHC for the Pageblocks dataset by generation.

pendent but roughly equally sized subsets, called folds, and applies the IS algorithm separately to each fold. Figure 3 shows that if CHC is applied in this manner then the fitness function plateaus well before 10,000 generations.

Given the observations in this section it is clear that 10,000 generations is an arbitrarily chosen stopping criterion. Additionally, a termination criterion that observes the convergence of the fitness function can identify promising stopping areas. In the following section we perform comparative experiments for 51 datasets where we compare the 3-NN classifier before and after IS to show the benefits of this new stopping criterion.

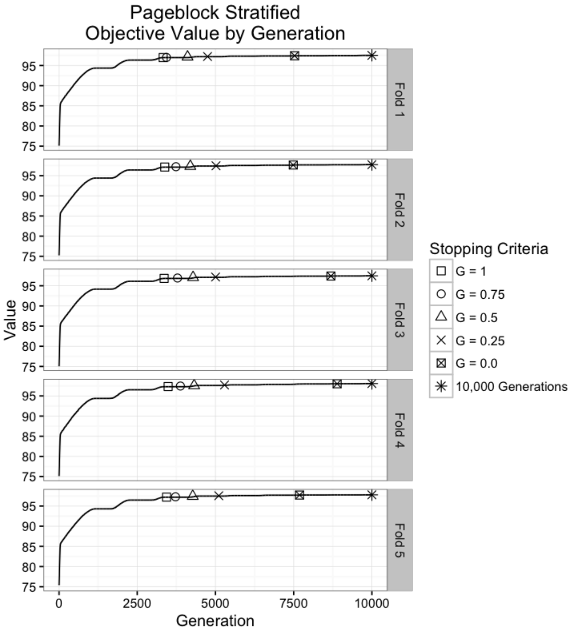

Fig. 3 Fitness of CHC for folds of the Pageblocks dataset by generation.

4 Experimental Study

In this section we perform comparative experiments for the datasets described in Table 5 (included at the end of the paper). All datasets in this paper are publicly available as is source code for the CHC and DROP3 IS algorithm [2] (granted without the modification for our stopping criterion). In the following subsections we introduce our experimental design and then an analysis of the results.

4.1 Experimental Design

To show the benefits of our stopping criterion we perform CHC on each dataset in Table 5 for the 3-NN classifier. CHC with our stopping criterion is terminated as described in Section 3 with $G = 0.0$, and called CHC_0. CHC with the 10,000 generation stopping criterion is called CHC_10K. Also computed for each dataset is DROP3, with $k = 3$, and the 3-NN classifier without IS. Predictive accuracy, dataset reduction, and computation times are estimated with 10-fold cross validation, where CHC_0, CHC_10K, DROP3, and 3-NN are applied to the same folds. Here predictive accuracy is taken as the average percentage of instances correctly labeled by the classifier for the withheld fold of the cross validation procedure. Additionally, CHC_10K and CHC_0 are applied to each fold three times with the results averaged for each fold (a total of 30 for each dataset). Finally, CHC_0, CHC_10K, and DROP3 are applied to the medium datasets using the stratified scaling method described in Section 3 and [4]. The number of folds for the stratified approach (not to be confused with the 10-fold cross validation procedure) is chosen as:

$$Folds = max\left\{2, \left\lfloor \frac{|TR|}{750} \right\rfloor \right\}$$

The three hypotheses listed below are tested using the Wilcoxon Signed Ranked test [9], with visualizations and additional analysis generated after the statistical tests. Also, hypothesis testing is done separately for small and medium datasets because of the applied scaling procedure.

1. The predictive accuracy obtained as a result of CHC_0 is equivalent to CHC_10K.
   $H_0 : Predictive\ Accuracy\ CHC\_10K - Predictive\ Accuracy\ CHC\_0 = 0$

2. The dataset reduction obtained as a result of CHC_0 is equivalent to CHC_10K.
   $H_0 : Reduction\ CHC\_10K - Reduction\ CHC\_0 = 0$

3. The computation time (in seconds) of CHC_0 is equivalent to CHC_10K.
   $H_0 : Time\ CHC\_10K - Time\ CHC\_0 = 0$

Experiments were executed with a 12 core machine possessing Dual Intel X5650 processors. Results were analyzed using the statistical programming language R and visualizations were generated using the R package ggplot2 [23].

4.2 Experimental Results

In this section we want to compare the performance of CHC_10K to CHC_0. Reference points from the DROP3 and the original 3-NN classifier are included to show the relative performance of CHC to other established methods.

Dataset Size	Hypothesis	Reject H_0	P-value
Small	1	No	1.0
Small	2	Yes	0.0
Small	3	Yes	0.0
Medium	1	No	0.1
Medium	2	Yes	0.0
Medium	3	Yes	0.0

Table 3 Results of Wilcoxon Signed Rank test.

Dataset Size	Method	Accuracy	Reduction	Time (S)
Small	3-NN	78.6	None	None
Small	CHC_10K	77.3	91.1	119
Small	CHC_0	77.3	90.6	64
Small	DROP3	76.1	90.7	1
Medium	3-NN	78.7	None	None
Medium	CHC_10K	75.4	90.9	1631
Medium	CHC_0	75.6	90.8	1415
Medium	DROP3	73.7	92.8	17

Table 4 Average accuracy, percent reduction of training dataset, and computation time of the IS method separated by classification method and dataset size, as estimated with 10-fold cross validation.

When considering predictive accuracy, Table 3, Hypothesis 1, shows that the CHC_10K and CHC_0 methods are not significantly different at the $p = 0.05$ level. However, in comparison to the original 3-NN classifier, Table 4 shows that the CHC methods decrease predictive accuracy on average. Still, it is observed that the average reduction in accuracy by CHC is not as great as with the DROP3 method. Additionally, Figure 4 shows there are times when CHC actually improves predictive accuracy, as with the Australian and Haberman datasets.

In regard to reduction, Table 3, Hypothesis 2, shows that CHC_10K leads to significantly better reduction than CHC_0. Still, the difference is less than a half percentage point on average. Surprisingly, DROP3 performs very well when only considering reduction, but it does sacrifice more predictive accuracy than the CHC methods. Figure 5 shows a dataset by dataset comparison of the two CHC methods and it is easy to see that in the majority of cases the reduction of CHC_0 is only slightly less than that of CHC_10K. Regardless, if the difference in reduction between CHC_10K and CHC_0 is too large, it is possible to further adapt the new stopping criterion to allow additional searching, or perhaps establish a termination criterion based on a reduction target.

Finally, Table 3, Hypothesis 3, shows that CHC_10K takes significantly more time to complete than CHC_0. Figure 6 shows the dataset by dataset comparison, and it can be seen that the new stopping criterion can greatly reduce the required computation time of IS. However, an interesting observation can be made when viewing the Chess and Splice datasets. The CHC_0 method actually took longer. For these datasets we can see that CHC_10K terminates too early because CHC_0 achieves both greater reduction and better predictive accuracy. Notably, Table 4 shows that DROP3 is a very fast method in comparison to CHC. We feel practitioners should weigh the importance of time required to perform IS and their required end performance when selecting an IS method.

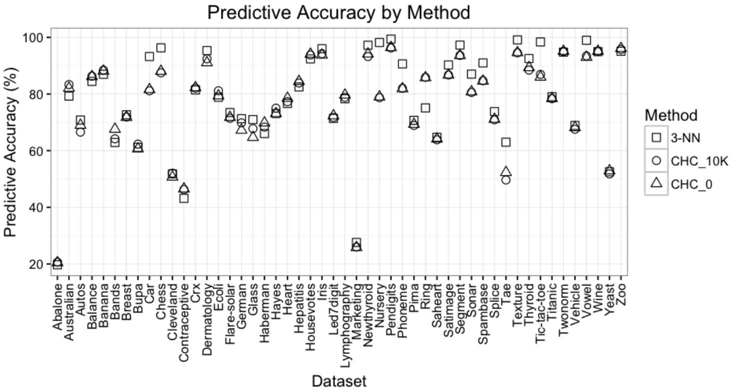

Fig. 4 Predictive accuracy of all datasets. Note that sometimes IS can be beneficial, other times detrimental. Users should weigh reduction/accuracy trade-offs for their specific applications.

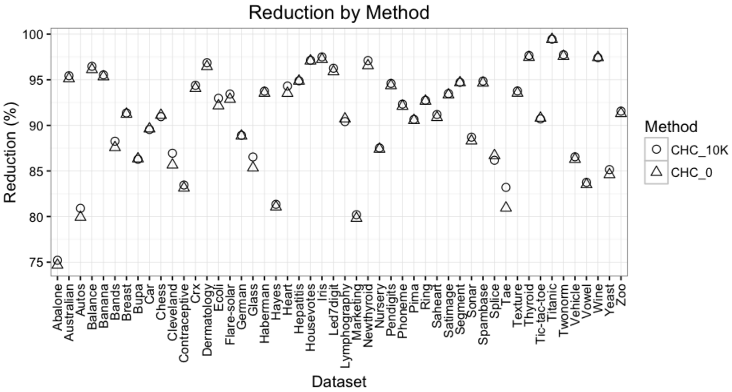

Fig. 5 Dataset reduction of all datasets.

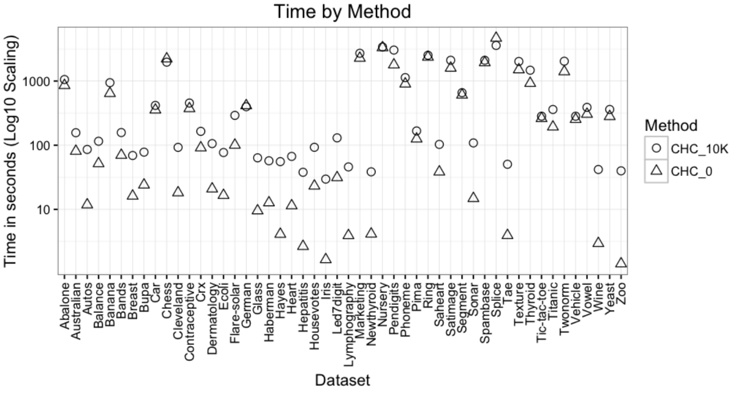

Fig. 6 Execution time of IS for all datasets.

5 Conclusion

In this work we proposed a new stopping criterion for the CHC Instance Selection algorithm. It was shown that the previous stopping criterion of 10,000 generations was too large for small datasets, and insufficient for medium datasets. Instead we proposed a criterion that terminates based on the convergence of the fitness function. Specifically, we terminate CHC if the fitness function is not markedly improved after a cataclysmic re-population in comparison to the previous cataclysmic re-population phase.

Experimental results for 51 datasets showed that our stopping criterion achieves similar accuracy to the old criterion, but with slightly less dataset reduction. On average, the new criterion results in 0.3 percentage points more data being selected. In regard to computation time, the new termination criterion reduces computation time on average by 55 seconds for small datasets, and by 216 seconds on average for medium datasets. This improvement, incorporated with scaling measures, will allow CHC to be more practically applied to large datasets. The result of which is greatly reduced storage requirements and classification times for instance based classifiers while maintaining acceptable classification accuracy.

Although experimental results show that our new stopping criterion can result in less reduction, it highlights the greater need for a stopping criterion that is based on the evolution of the solution.. For example, with the Splice dataset the new criterion took longer to terminate than the original method, but greater reduction and increased accuracy was achieved. Alternatively, for the Chess dataset, the original method took longer to complete, but little (if any) benefit is seen in reduction or accuracy. Our method is one possible stopping criterion, but others could be derived for any particular objective.

Future work should consider additional stopping criteria for the CHC Instance Selection algorithm, such as, a threshold that terminates the search when a certain level of reduction is achieved. Also, an investigation into the trade-offs between dataset reduction and predictive accuracy should be conducted. Practitioners will likely want to determine apriori at what rate predictive accuracy is harmed by performing IS for a dataset.

	Dataset	Instances	Attributes	Classes
Small	Australian	690	14	2
	Autos	205	25	6
	Balance	625	4	3
	Bands	538	19	2
	Breast	286	8	2
	Bupa	345	6	2
	Cleveland	303	13	5
	Crx	690	15	2
	Dermatology	366	34	6
	Ecoli	335	7	8
	German	1000	20	2
	Glass	214	9	6
	Haberman	306	3	2
	Hayes	160	4	3
	Heart	270	13	2
	Hepatitis	155	19	2
	Housevotes	435	16	2
	Iris	150	4	2
	Led7digit	500	7	10
	Lymphography	148	18	4
	Newthyroid	215	5	3
	Pima	768	8	2
	Saheart	462	9	2
	Sonar	208	60	2
	Tae	151	5	3
	Tic-tac-toe	958	9	2
	Vehicle	846	18	4
	Vowel	990	13	2
	Wine	178	13	3
	Zoo	101	16	7
Medium	Abalone	4174	8	28
	Banana	5300	2	2
	Car	1728	6	4
	Chess	3196	36	2
	Coil2000	9822	85	2
	Contraceptive	1473	9	3
	Flare-solar	1066	11	6
	Marketing	8993	13	9
	Nursery	12960	8	5
	Pendigits	10992	16	2
	Ring	7400	20	2
	Phoneme	5404	5	2
	Satimage	6435	36	6
	Segment	2310	19	7
	Spambase	4597	57	2
	Splice	3190	60	3
	Texture	5500	40	11
	Thyroid	7200	21	3
	Titanic	2201	3	2
	Twonorm	7400	20	2
	Yeast	1484	8	10

Table 5 Experimental datasets, where “Small” datasets are those with 1,000 or fewer instances.

References

1. D. W. AHA, D. KIBLER, AND M. K. ALBERT, *Instance-Based Learning Algorithms*, Machine learning, 6 (1991), pp. 37–66.
2. J. ALCALÁ-FDEZ, A. FERNÁNDEZ, J. LUENGO, J. DERRAC, S. GARCÍA, L. SÁNCHEZ, AND F. HERRERA, *KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework*, Journal of Multiple-Valued Logic and Soft Computing, 17 (2011), pp. 255–287.
3. CANO, F. HERRERA, AND M. LOZANO, *Evolutionary stratified training set selection for extracting classification rules with trade off precision-interpretability*, Data and Knowledge Engineering, 60 (2007), pp. 90–108.
4. J. CANO, F. HERRERA, AND M. LOZANO, *Using evolutionary algorithms as instance selection for data reduction in KDD: an experimental study*, IEEE Transactions on Evolutionary Computation, 7 (2003), pp. 561–575.
5. J. R. CANO, F. HERRERA, AND M. LOZANO, *On the combination of evolutionary algorithms and stratified strategies for training set selection in data mining*, Applied Soft Computing, 6 (2006), pp. 323–332.
6. T. COVER AND P. HART, *Nearest neighbor pattern classification*, IEEE Transactions on Information Theory, 13 (1967), pp. 21–27.
7. A. DE HARO-GARCÍA AND N. GARCÍA-PEDRAJAS, *A divide-and-conquer recursive approach for scaling up instance selection algorithms*, Data Mining and Knowledge Discovery, 18 (2009), pp. 392–418.
8. A. DE HARO-GARCÍA, N. GARCÍA-PEDRAJAS, AND J. A. R. DEL CASTILLO, *Large scale instance selection by means of federal instance selection*, Data and Knowledge Engineering, 75 (2012), pp. 58–77.
9. J. DEMŠAR AND J. DEMŠAR, *Statistical Comparisons of Classifiers over Multiple Data Sets*, The Journal of Machine Learning Research, 7 (2006), pp. 1–30.
10. ESHELMAN, *The CHC Adaptive Search Algorithm. Foundations of Genetic Algorithms*, Morgan-Kaufmann, 1991.
11. S. GARCÍA, J. R. CANO, A. FERNANDEZ, AND F. HERRERA, *A Proposal of Evolutionary Prototype Selection for Class Imbalance Problems*, (2006), pp. 1415–1423.
12. S. GARCÍA, J. DERRAC, I. TRIGUERO, C. J. CARMONA, AND F. HERRERA, *Evolutionary-based selection of generalized instances for imbalanced classification*, Knowledge-Based Systems, 25 (2012), pp. 3–12.
13. S. GARCÍA, A. FERNANDEZ, AND F. HERRERA, *Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems*, Applied Soft Computing Journal, 9 (2009), pp. 1304–1314.
14. S. GARCÍA, J. LUENGO, AND F. HERRERA, *Data Preprocessing in Data Mining*, vol. 72 of Intelligent Systems Reference Library, Springer International Publishing, Cham, 2015.
15. N. GARCÍA-PEDRAJAS, *Evolutionary computation for training set selection*, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 1 (2011), pp. 512–523.
16. N. GARCÍA-PEDRAJAS, J. PEEZ-RODRÍGUEZ, AND A. DE HARO-GARCIÁ, *OligoIS: Scalable instance selection for class-imbalanced data sets*, IEEE Transactions on Cybernetics, 43 (2013), pp. 332–346.
17. P. HART, *The condensed nearest neighbor rule (Corresp.)*, IEEE Transactions on Information Theory, 14 (1968), pp. 1966–1967.
18. J. A. OLVERA-LOPEZ, J. A. CARRASCO-OCHOA, J. F. MARTINEZ-TRINIDAD, AND J. KITTLER, *A review of instance selection methods*, Artificial Intelligence Review, 34 (2010), pp. 133–143.
19. C. PASSINI AND M. LUIZA, *A Strategy for Training Set Selection in Text Classification Problems.*, International Journal of Advanced Computer Science & Applications, 4 (2013), pp. 54–60.
20. G. RITTER, H. WOODRUFF, S. LOWRY, AND T. ISENHOUR, *An algorithm for a selective nearest neighbor decision rule (Corresp.)*, IEEE Transactions on Information Theory, 21 (1975).

21. M. Safe, J. Carballido, I. Ponzoni, and N. Brignole, *On Stopping Criteria for Genetic Algorithms*, Advances in Artificial Intelligence, 17 (2004), pp. 405–413.
22. M. Sebban, R. Nock, J. H. Chauchat, and R. Rakotomalala, *Impact of learning set quality and size on decision tree performances*, International Journal of Computers, Systems and Signals, 1 (2000), pp. 85–105.
23. H. Wickham, *ggplot2: elegant graphics for data analysis*, 2009.
24. D. L. Wilson, *Asymptotic Properties of Nearest Neighbor Rules Using Edited Data*, IEEE Transactions on Systems, Man, and Cybernetics, 2 (1972), pp. 408–421.
25. D. R. Wilson and T. R. Martinez, *Reduction techniques for instance-based learning algorithms*, Machine Learning, 38 (2000), pp. 257–286.
26. X. Zhu and X. Wu, *Scalable representative instance selection and ranking*, Proceedings - International Conference on Pattern Recognition, 3 (2006), pp. 352–355.