Lecture Notes In Artificial Intelligence, vol. 8468, pp. 40–51, ICAISC 2014

The original publication is available at www.springerlink.com

# Bagging of Instance Selection Algorithms

Marcin Blachnik[1] and Mirosław Kordos[2]

Silesian University of Technology, Department of Management and Informatics,
Katowice, Krasińskiego 8, Poland; marcin.blachnik@polsl.pl
University of Bielsko-Biala, Department of Mathematics and Computer Science,
Bielsko-Biała, Willowa 2, Poland; mkordos@ath.bielsko.pl

**Abstract.** The paper presents bagging ensembles of instance selection algorithms. We use bagging to improve the results of single instance selection algorithms in terms of compression and classification accuracy or the mean squared error in case of regression problems. The examined instance selection algorithms for classification are ENN, CNN, RNG and GE and for regression are the developed by us Generalized CNN and Generalized ENN algorithms. Results of the comparative experimental study performed using different configurations on several datasets shows that this approach allowed for significant improvement, especially in terms of data compression.

## 1 Introduction

### 1.1 Motivation of this Work

Reducing the number of instances in the training dataset has several advantages:

1. Removing outliers reduce noise in the dataset, thus improving the predictive model accuracy.
2. Discarding instances that are too similar to each other prevents overfiting during the training.
3. The above two approaches can be joined together to obtain the benefits of both.
4. The learning a classifier on a smaller dataset is faster. Although reducing the dataset size also takes some time, it frequently can be done only once, before attempting various models with various parameters to match the best model for the problem.
5. While using lazy-learning algorithms, as k-NN, reducing the dataset size also reduces the prediction time.
6. Instance selection can also be used as prototype selection, e.g. in prototype rule-based systems.

However, there is still a field of improvement, because the following issues haven't been researched sufficiently by now:

1. Most research on instance selection, which has been done so far, refers to classification problems, while instance selection for regression tasks is very important in many practical applications.

2. In the case of classifications the instance selection algorithms usually do not have robust tunable parameters to make them more or less sensible to outliers and instances very similar to each other or to balance the accuracy vs. data size tradeoff.
3. The decision which of two close instances should be removed frequently depends on the order in which the instances are considered, which can lead to suboptimal decisions.

In this work we want to address the above list of issues.

### 1.2 Instance Selection in Classification Problems

In that area many research have been done using the k-nearest neighbor algorithm (k-NN) for instance selection in classification tasks. The early research in that area lead to the Condensed Nearest Neighbor rule (CNN) [3] and Edited Nearest Neighbor rule (ENN) [13]. These basic algorithms were further extended leading to more complex ones like Drop1-5 [13], IB3, Gabriel Editing (GE) and Relative Neighborhood Graph Editing (RNGE), Iterative Case Filtering (ICF), ENRBF2, ELH, ELGrow and Explore [7]. Instead of directly selecting instances from the training data an interesting approach for training k-NN classifier was proposed by Kunheva, where preselected instances were relabelled, such that each instance was assigned to all class labels with appropriate weights describing the support for given label. A large survey including almost 70 different algorithms of instance selection for classification tasks can be found in [7].

### 1.3 Instance Selection in Regression Problems

The instance selection issue for regression tasks is much more complex. The reason is that in classification tasks only the boundaries between classes must be precisely determined, while in regression tasks the output value must be properly calculated at each point of the input space. Moreover, the decision in classification tasks is frequently binary or there are at most several different classes, while in regression tasks, the output of the system is continuous, so there is an unlimited number of possible values predicted by the system. That causes that the dataset compression obtained by instance selection can be much higher in classification than in non-linear regression problems. Moreover, the decision about rejection of a given vector in classification tasks can be made based on right or wrong classification of the vector by some algorithm. In regression problems, rather some threshold defining the difference between the predicted and the actual value of the vector output should be set and determining the threshold is another issue that does not exist in classification tasks. Because of the challenges, there were very few approaches in the literature to instance selection for regression problems. Moreover, the approaches were usually not verified on real-world datasets. Zhang [15] presented a method to select the input vectors while calculating the output with k-NN. Tolvi [8] presented a genetic algorithm to perform feature and instance selection for linear regression models. In their works Guillen et al. [2] discussed the concept of mutual information used for selection of prototypes in regression problems.

## 2 Instance Selection Algorithms Used in Experiments

### 2.1 ENN and CNN Instance Selection Methods

The CNN (Condensed Nearest Neighbor) algorithm was proposed by Hart [3]. For classification problems, CNN condenses on the average the number of vectors three times. CNN used for classification works in the following way: the algorithm starts with only one randomly chosen instance from the original dataset T. And this instance is added to the new dataset P. Then each remaining instance from T is classified with the k-NN algorithm, using the k nearest neighbors from the dataset P. If the classification is correct, then the instance is not added to the final dataset P. If the classification is wrong - the instance is added to P. Thus, the purpose of CNN is to reject these instances, which do not bring any additional information into the classification process.

The ENN (Edited Nearest Neighbor) algorithm was created by Wilson [13]. The main idea of ENN is to remove given instance if its class is different than the majority class of its neighbors, thus ENN works as a noise filter. ENN starts from the entire original training set T. Each instance, which is correctly classified by its k nearest neighbors is added to the new dataset P and each instance wrongly classified is not added. Several variants of ENN exist. Repeated ENN, proposed by Wilson, where the process of ENN is iteratively repeated as long as there are any instances wrongly classified.

### 2.2 GenENN and GenCNN: ENN and CNN for Regression Problems

The first step to modify the CNN and ENN algorithms to enable using them for regression task is to replace the wrong/correct classification decision with a distance measure and a similarity threshold, to decide if the examined vector can be considered as similar to its neighbors or not. For that purpose we use Euclidean measure and a threshold $\theta$, which express the maximum difference between the output values of two vectors to consider them similar. Using $\theta$ proportional to the standard deviation of several nearest neighbors of the vector $\mathbf{x}_i$ reflects the speed of changes of the output around $\mathbf{x}_i$ and allows adjusting the threshold to that local landscape, what, as the experiments showed, allows for better compression of the dataset. We also allow the algorithm used to predict the output $Y(\mathbf{x}_i)$ to be any classification or regression model, as k-NN, MLP, decision tree and others, what in many cases allows for result improvement [5].

In the pseudo-code $\mathbf{T}$ is the training dataset, $\mathbf{P}$ is the set of selected prototypes, $\mathbf{x}_i$ is the i-th vector, $m$ is the number of vectors in the dataset ,$Y(\mathbf{x}_i)$ is the real output value of vector $\mathbf{x}_i$, $\bar{Y}(\mathbf{x}_i)$ is the predicted output of vector $\mathbf{x}_i$, $S$ is the set of nearest neighbors of vector $\mathbf{x}_i$, $Model$ is the algorithm, which is trained on dataset $\mathbf{T}$ and vector $\mathbf{x}$ is used as a test sample, for which the $Y(\mathbf{x}_i)$ is predicted (in the experiments with regression datasets a k-NN with $k = 9$ was used), k-NN is the k-NN algorithm returning the subset $S$ of $k$ closest neighbors to $\mathbf{x}_i$. $\Theta$ is the threshold of acceptance/rejection of the vector as a prototype, thus $\Theta$ expresses the maximum difference between the output values of two vectors to consider them similar. $\alpha$ is a certain constant coefficient and $std\left(Y\left(\mathbf{X}_S\right)\right)$ is the standard deviation of the outputs of the vectors in $S$. Using $\Theta$ proportional to the standard deviation of $k$ nearest neighbors of the vector $\mathbf{x}_i$ reflects the speed of changes of the output around $\mathbf{x}_i$ and allows adjusting the threshold to that local landscape, what, as our past experiments showed [5], allows for higher compression of the dataset.

**Algorithm 1** RegENN algorithm

```
Require: $\mathbf{T}$
  $m \leftarrow sizeof(\mathbf{T})$;
  for $i = 1 \ldots m$ do
    $\bar{Y}(\mathbf{x}_i)$ =Model($(\mathbf{T} \setminus \mathbf{x}_i), \mathbf{x}_i$);
    $S \leftarrow$ k-NN($\mathbf{T}, \mathbf{x}_i$)
    $\theta = \alpha \cdot std\,(Y\,(\mathbf{X}_S))$
    if $\left|Y(\mathbf{x}_i) - \bar{Y}(\mathbf{x}_i)\right| > \theta$ then
      $\mathbf{T} \leftarrow \mathbf{T} \setminus \mathbf{x}_i$
    end if
  end for
  $\mathbf{P} \leftarrow \mathbf{T}$
  return $\mathbf{P}$
```

**Algorithm 2** RegCNN algorithm

```
Require: $\mathbf{T}$
  $m \leftarrow sizeof(\mathbf{T})$
  $\mathbf{P} = \emptyset$
  $\mathbf{P} \leftarrow \mathbf{P} \cup \mathbf{x}_1$;
  for $i = 2 \ldots m$ do
    $\bar{Y}(\mathbf{x}_i)$ =Model($\mathbf{P}, \mathbf{x}_i$)
    $S \leftarrow$ k-NN($\mathbf{T}, \mathbf{x}_i$)
    $\theta = \alpha \cdot std\,(Y\,(\mathbf{X}_S))$
    if $\left|Y(\mathbf{x}_i) - \bar{Y}(\mathbf{x}_i)\right| > \theta$ then
      $\mathbf{P} \leftarrow \mathbf{P} \cup \mathbf{x}_i$;
      $\mathbf{T} \leftarrow \mathbf{T} \setminus \mathbf{x}_i$
    end if
  end for
  return $\mathbf{P}$
```

### 2.3 GE and RNGE

Gabriel Editing (GE) and Relative Neighborhood Graph Editing (RNGE) are two algorithms based on graph theory, described by Bhattacharya [1]. The complexity of them is $O(n^3)$.

A Gabriel Graph is a graph with vertex set $\mathbf{T}$ obtained by connecting any two points any points $\mathbf{p}_a$ and $\mathbf{p}_b$ in $\mathbf{T}$, which are considered the nearest neighbors, that is if the circle which diameter is a line segment connecting the points $\mathbf{p}_a$ and $\mathbf{p}_b$ does not contain inside any other elements denoted as $\mathbf{p}_c$ of $\mathbf{T}$. In other words for every triple of points the formula:

$$\underset{a \neq b \neq c}{\forall} D^2(\mathbf{p}_a, \mathbf{p}_b) > D^2(\mathbf{p}_a, \mathbf{p}_c) + D^2(\mathbf{p}_b, \mathbf{p}_c) \tag{1}$$

is evaluated. If the inequality is fulfilled vectors $\mathbf{p}_a$ and $\mathbf{p}_b$ are marked as neighbors.

A Relative Neighborhood Graph (RNG) is very similar to GE algorithm. The main difference is in the formula determining graph neighbors:

$$\underset{a \neq b \neq c}{\forall} D(\mathbf{p}_a, \mathbf{p}_b) \geq \max(D(\mathbf{p}_a, \mathbf{p}_c), D(\mathbf{p}_b, \mathbf{p}_c)) \tag{2}$$

In both of this algorithms the remaining instances are thous which fulfills the above formulas and belong to opposite classes.

**Algorithm 3** GE and RNGE algorithm

```
Require: $\mathbf{T}$
  Construct the Gabriel Graph (or Relative Neighborhood Graph) of the original set $\mathbf{T}$.
  Visit each node of $\mathbf{T}$ and mark the visited node if all its Gabriel neighbors are not from the
  same class as the node visited (or differ by more than $\theta$ in the case of regression).
  Discard all data points (all the nodes), which are not marked.
  The marked data points are the selected instances (the Gabriel or the RNG edited set $\mathbf{P}$).
  return $\mathbf{P}$
```

## 3 Bagging Method for Instance Selection

For last decade or even more ensemble learning [9] proved to be one of the most profitable directions in machine learning. This approach is often called meta-learning and confirmed its quality in various data mining challenges [11]. In this area many directions have been proposed but the general idea is based on replacing a single expert by a group of experts voting non necessary with equal rights for the best solution. Ensemble learning includes solutions such as voting, stacking, bagging or boosting, etc. Recently these kind of methods were used for ensemble of final prediction models, but there is a little work on using such an approach for data preprocessing such us instance selection. In this paper we investigate application of bagging to improve the quality of instance selection.

In case of classical applications, bagging is based on building several individual prediction models on subset of original training instances, where these data subsets are sampled with replacement from the initial training dataset. An example of such an algorithm is RandomForest [10] where each decision tree is built on a subset of features and a subset of training samples. When making the prediction each of the component models is voting for the final output (in case of classification) or the individual predictions are averaged (in case of regression).

The idea of bagging can be also adapted for the instance selection. In that case each run of the instance selection algorithm provides binary weights denoting absence or presence of each of the training samples. The collected votes for each training instance are combined and averaged, what allows to assign a single new weight attribute representing the importance of each training sample.

Finally to perform the final instance selection an *acceptance threshold* is defined that determines which instances will be included in the final training set.

This approach has several issues. The instance selection method can be simply divided into the outlier eliminators such as ENN algorithm and redundancy filters such as CNN algorithm. The first group usually allows to remove a small subset of instances what leads to small compression, and rather dense final set, while the redundancy removal methods has large compression and usually small, though sparse final subset. The second group of methods may be very problematic in case of bagging applications, especially for large datasets because for sparse datasets the probability of selecting the same vectors highly decreases, determining the use of small values of *acceptance threshold*, and reducing the compression coefficient.

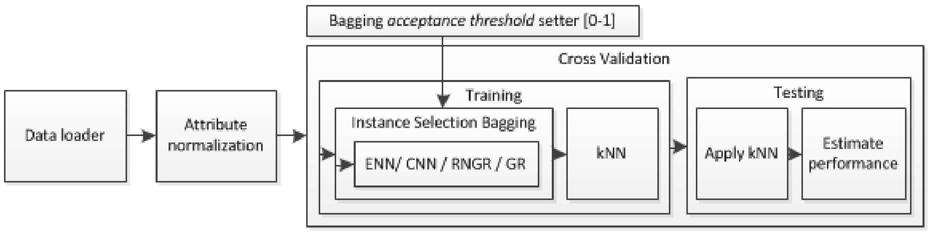


**Fig. 1.** The experimental evaluation process

## 4 Eperimental Evaluation

### 4.1 Experimental Environment

We implemented the instance selection algorithms in Java as RapidMiner Extensions and used RapidMiner [6] for the whole process. The source codes, executable files and datasets used in the experiments can be downloaded from [16].

In our experiments we tested four different instance selection algorithms for classification: ENN, CNN, RNGE, GE and two methods for regression: Generalized ENN and Generalized CNN, which were developed from ENN and CNN to extend their functionality so that they cover any tasks where the cost function can be defined (classification, regression). The parameters of the instance selection methods were set to the default values. In the case of classification we used 1-NN as the inner model of these algorithms and as the classifier. In the case of regression we used 9-NN as the inner model and 9-NN as the predictor.

We performed the experiments on six most popular classification and four regression datasets. The classification datasets used in the experiments were obtained from the UCI Machine Learning Repository [12]: heart disease, ionosphere, Pima Indian diabetes, sonar, vehicle and Wisconsin breast cancer. For the regression problems two datasets were also obtained form the UCI repository: Concrete Compression Strength (7 attributes, 1030 instances), Crime and Communities (7 attr., 320 inst.). Two datasest (SteelC14 and SteelC14-noise: 18 attr., 2382 inst.) depict the steel production process with the task to predict the amount of carbon that must be added to the liquid steel in to obtain the desired steel properties.

For the purpose of the experiment we defined a RapidMiner process, which is available from [16]. The scheme of the testing environment is presented in figure (4.1) The process starts form loading the data and attribute normalization or standardization (in case of regression we did standardization). After it, according to the *acceptance threshold* settings and selection of instance reduction method the 10 fold cross-validation was performed, where in each validation first the bagging algorithm was executed which wraps the instance reduction method. In our experiments the instance reduction method was executed 15 times on 80% of instances from the training set sampled without replacement. After bagging, the selected instances were used to train the k-NN algorithm, which was applied on the test set. Finally the average performance (classification accuracy in case of classification or MSE in case of regression) and compression were evaluated. In our case the compression $C_x$ is defined as

$$C_x = 1 - \frac{number\_of\_instances\_after\_selection}{number\_of\_instances\_before\_selection} \quad (3)$$

and it tends to 1 if less instances remain in the training set, and tends to 0 if no instance reduction is possible.

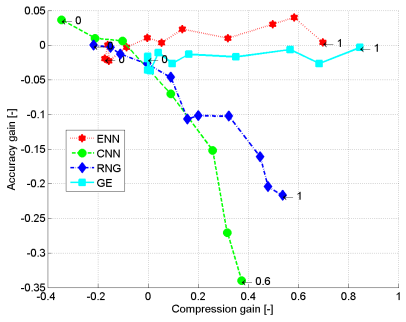
### 4.2 Simulation Results

The simulation results are presented in tables in appendix. Typically, results of instance selection methods are represented in a form of accuracy-compression plots. This represents both of the goals of instance selection; rejection of unneeded instances and improvement of the model accuracy. The training dataset compression shows rejection of redundant and thus not informative instances as well as rejection of the outliers, which do not fit in the model. In other words the plots show relation between the level of compression and the instance selection influence on the accuracy of the final prediction system. However, in our case to perform the comparison between bagging-based instance selection and a single instance selector we draw accuracy-compression gain plots, where the horizontal axis expresses

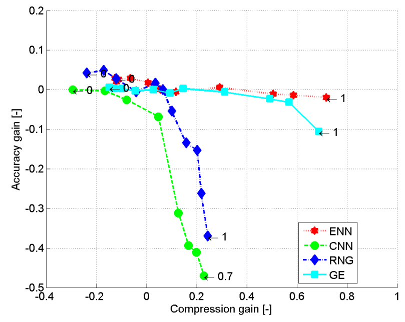
$accuracy\_of\_bagging\_method - accuracy\_of\_single\_method$
and the vertical axis expresses
$compression\_of\_bagging\_method - compression\_of\_single\_method$
such that values equal 0 represent no gain, and values $\geq 0$ represent the level of improvement, while values $\leq 0$ the level of deterioration. The gain plots are presented in figures 4.2 and 4.2.

## 5 Conclusions

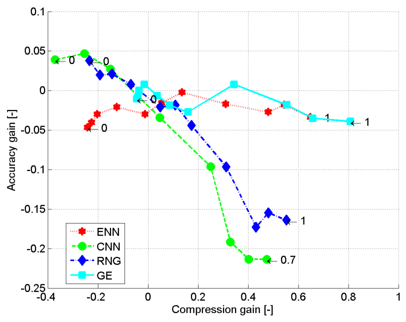
The obtained results showed that using the bagging ensembles is beneficial when the proper *acceptance threshold* is set. Generally the highest gain was obtained for the ENN and GE algorithms. In that case the bagging allowed increasing the compression level of up to 80% without any significant accuracy drop. Bagging also allowed improving the classification accuracy of up to 5% in almost all cases however reducing the compression. For the CNN and RNGE algorithms high accuracy drop may be observed for increased *acceptance threshold* value. These results can be explained by the high initial compression ratio. As it was already mentioned, because CNN initially boast very high compression (what can't be observed on the gain plot), it becomes less likely that the same instance will be selected many times, which reduces the possible level of applicable *acceptance threshold*. Similar results were also obtained for the regression problems, where too high *acceptance threshold* decreases the performance measured by RMSE ( (4.2)). The positive value of *RMSE-gain* indicates reduction of the absolute RMSE value.



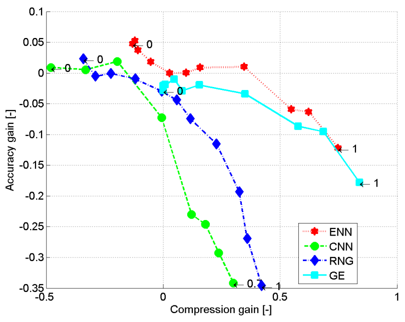
(a) Heart Disease



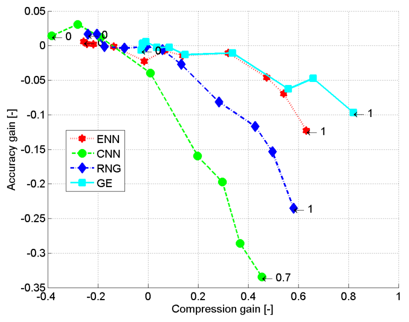
(b) Ionosphere



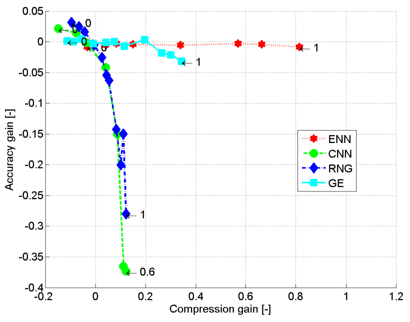
(c) Diabetes



(d) Sonar



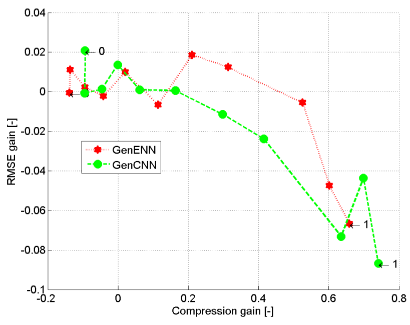
(e) Vehicle



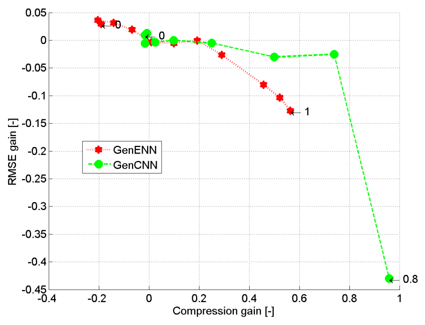
(f) WBC

**Fig. 2.** Comparison of relative improvement of accuracy and compression of the bagging ensemble of instance selection methods vs. the original method without bagging as a function of the *acceptance threshold* for classification tasks

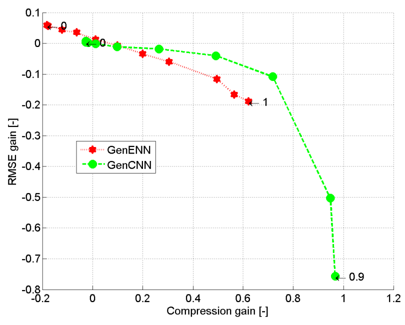
The final conclusion is that bagging ensembles of instance selection algorithms can perform better that the single algorithms (what was especially spectacular in the compression gain of ENN bagging ensembles). On the other hand frequently the CNN bagging ensembles allowed for better accuracy gain of up to 5% but with carefully tuned *acceptance threshold* taking small values. Thus a good direction of the future research



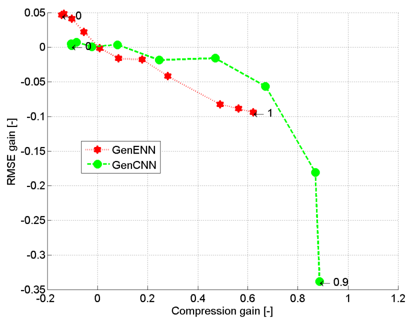
(a) Concrete



(b) Crime



(c) SteelC14



(d) SteelC14 + noise

**Fig. 3.** Comparison of relative improvement of RMSE and compression of the bagging ensemble of instance selection methods vs. the original method without bagging as a function of the *acceptance threshold* for regression tasks

would be to build an other ensemble of instance selection methods. One of the possibilities are voting methods which may replace the classical approach based on serial processing of instance selection for example including the benefits of both CNN and ENN and thus attempt to improve significantly both: compression gain and accuracy gain.

## References

1. Bhattacharya, B.K., Poulsen, R.S., Toussaint, G.T.: Application of proximity graphs to editing nearest neighbor decision rule. In: International Symposium on Information Theory, Santa Monica (1981)
2. Guillen, A.: Applying mutual information for prototype or instance selection in regression problems. In: ESANN 2009 (2009)
3. Hart, P.: The condensed nearest neighbor rule (corresp.). Information Theory, IEEE Transactions on 14(3), 515–516 (1968)
4. Huber, P.J.: Robust Statistics. Wiley Series in Probability and Statistics, Wiley-Interscience (1981)
5. Kordos, M., Białka, S., Blachnik, M.: Instance selection in logical rule extraction for regression problems. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh,

L., Zurada, J. (eds.) Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, vol. 7895, pp. 167–175. Springer Berlin Heidelberg (2013)
6. http://www.rapidminer.com
7. Salvador, G., Derrac, J., Ramon, C.: Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE Transactions on Pattern Analysis and Machine Intelligence 34, 417–435 (2012)
8. Tolvi, J.: Genetic algorithms for outlier detection and variable selection in linear regression models. Soft Computing 8, 527–533 (2004)
9. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley, (2004)
10. Breiman,L: Random Forests. Machine Learning 45 (1): 5–32, (2001)
11. Jankowski N., Gr?bczewski K.. Handwritten digit recognition — road to contest victory. In IEEE Symposium Series on Computational Intelligence, IEEE Press, 491–498, (2007)
12. Merz, C., Murphy, P.: Uci repository of machine learning databases (2013), `http://www.ics.uci.edu/mlearn/MLRepository.html`
13. Wilson, D.L.: Asymptotic properties of nearest neighbor rules using edited data. Systems, Man and Cybernetics, IEEE Transactions on SMC-2(3), 408–421 (1972)
14. Wilson, D., Martinez, T.: Reduction techniques for instance-based learning algorithms. Machine Learning, Vol. 38, 251-268 (2000)
15. Zhang, J.: Intelligent selection of instances for prediction functions in lazy learning algorithms. Artifcial Intelligence Review 11, 175–191 (1997)
16. source code and datasets used in the paper http://code.google.com/p/instance-selection-2014/

## Appendix

**Table 1.** Classification accuracy and instance selection compression obtained in the experiments. Threshold denoted as "-" represent normal result without bagging ensemble

| Dataset | thr | Accuracy | Compress | Accuracy | Compress | Accuracy | Compress | Accuracy | Compress |
|---|---|---|---|---|---|---|---|---|---|
| - | | ENN | | CNN | | RNGE | | GE | |
| Sonar | - | 0.81±0.86 | 0.07±0.01 | 0.86±0.31 | 0.08±0.01 | 0.86±0.46 | 0.08±0.02 | 0.89±1.00 | 0.09±0.00 |
| | 0.00 | 0.86±0.99 | 0.07±0.01 | 0.87±0.79 | 0.08±0.03 | 0.88±0.80 | 0.06±0.02 | 0.86±1.00 | 0.08±0.00 |
| | 0.10 | 0.86±0.98 | 0.06±0.00 | 0.87±0.64 | 0.08±0.02 | 0.85±0.75 | 0.06±0.03 | 0.87±1.00 | 0.08±0.00 |
| | 0.20 | 0.85±0.96 | 0.09±0.01 | 0.88±0.51 | 0.05±0.02 | 0.86±0.68 | 0.07±0.02 | 0.87±1.00 | 0.07±0.00 |
| | 0.30 | 0.83±0.91 | 0.10±0.01 | 0.79±0.32 | 0.12±0.02 | 0.85±0.58 | 0.08±0.02 | 0.87±0.99 | 0.08±0.01 |
| | 0.40 | 0.81±0.83 | 0.07±0.02 | 0.63±0.19 | 0.10±0.02 | 0.83±0.47 | 0.06±0.01 | 0.88±0.95 | 0.08±0.01 |
| | 0.50 | 0.81±0.76 | 0.08±0.03 | 0.61±0.13 | 0.09±0.01 | 0.81±0.40 | 0.03±0.02 | 0.86±0.92 | 0.08±0.02 |
| | 0.60 | 0.82±0.70 | 0.09±0.02 | 0.57±0.07 | 0.10±0.02 | 0.78±0.34 | 0.12±0.02 | 0.87±0.85 | 0.04±0.02 |
| | 0.70 | 0.82±0.51 | 0.08±0.02 | 0.52±0.01 | 0.08±0.00 | 0.74±0.23 | 0.08±0.02 | 0.86±0.65 | 0.05±0.02 |
| | 0.80 | 0.75±0.31 | 0.10±0.01 | ± | ± | 0.66±0.13 | 0.10±0.01 | 0.80±0.42 | 0.08±0.02 |
| | 0.90 | 0.75±0.23 | 0.08±0.02 | ± | ± | 0.59±0.10 | 0.10±0.02 | 0.79±0.32 | 0.09±0.02 |
| | 1.00 | 0.69±0.11 | 0.10±0.02 | ± | ± | 0.51±0.04 | 0.11±0.01 | 0.71±0.16 | 0.08±0.02 |
| Ionosphere | - | 0.84±0.84 | 0.06±0.01 | 0.87±0.24 | 0.04±0.01 | 0.81±0.26 | 0.03±0.02 | 0.87±0.79 | 0.04±0.01 |
| | 0.00 | 0.87±0.97 | 0.04±0.01 | 0.87±0.54 | 0.05±0.01 | 0.86±0.50 | 0.05±0.01 | 0.87±0.94 | 0.04±0.01 |
| | 0.10 | 0.86±0.96 | 0.05±0.01 | 0.87±0.41 | 0.06±0.02 | 0.86±0.43 | 0.06±0.01 | 0.87±0.92 | 0.05±0.01 |
| | 0.20 | 0.86±0.96 | 0.06±0.01 | 0.85±0.32 | 0.07±0.01 | 0.84±0.38 | 0.06±0.02 | 0.87±0.89 | 0.04±0.01 |
| | 0.30 | 0.87±0.91 | 0.03±0.01 | 0.80±0.19 | 0.07±0.02 | 0.81±0.30 | 0.07±0.01 | 0.87±0.84 | 0.03±0.01 |
| | 0.40 | 0.85±0.84 | 0.06±0.01 | 0.56±0.11 | 0.11±0.01 | 0.83±0.23 | 0.06±0.02 | 0.87±0.77 | 0.05±0.01 |
| | 0.50 | 0.84±0.79 | 0.06±0.02 | 0.48±0.07 | 0.18±0.01 | 0.82±0.20 | 0.07±0.01 | 0.86±0.70 | 0.03±0.01 |
| | 0.60 | 0.83±0.72 | 0.06±0.01 | 0.46±0.04 | 0.17±0.01 | 0.76±0.16 | 0.03±0.01 | 0.87±0.65 | 0.04±0.02 |
| | 0.70 | 0.84±0.55 | 0.07±0.02 | 0.40±0.01 | 0.11±0.00 | 0.68±0.10 | 0.11±0.01 | 0.86±0.48 | 0.04±0.02 |
| | 0.80 | 0.83±0.34 | 0.06±0.02 | ± | ± | 0.66±0.06 | 0.11±0.01 | 0.85±0.30 | 0.06±0.02 |
| | 0.90 | 0.82±0.26 | 0.04±0.02 | ± | ± | 0.55±0.04 | 0.12±0.01 | 0.84±0.23 | 0.08±0.01 |
| | 1.00 | 0.82±0.12 | 0.10±0.01 | ± | ± | 0.44±0.02 | 0.15±0.01 | 0.76±0.11 | 0.06±0.02 |

Continued on next page

**Table 2 – continued from previous page**

| Dataset | thr | Accuracy | Compress | Accuracy | Compress | Accuracy | Compress | Accuracy | Compress |
|---|---|---|---|---|---|---|---|---|---|
| - | | ENN | | CNN | | RNGE | | GE | |
| Wisconsin breast cancer | - | 0.97±0.96 | 0.02±0.00 | 0.92±0.14 | 0.02±0.01 | 0.91±0.13 | 0.03±0.01 | 0.95±0.38 | 0.03±0.01 |
| | 0.00 | 0.96±0.99 | 0.02±0.00 | 0.94±0.29 | 0.02±0.02 | 0.94±0.22 | 0.03±0.01 | 0.95±0.50 | 0.03±0.01 |
| | 0.10 | 0.96±0.99 | 0.02±0.00 | 0.94±0.21 | 0.02±0.01 | 0.94±0.19 | 0.03±0.01 | 0.95±0.47 | 0.03±0.01 |
| | 0.20 | 0.96±0.99 | 0.02±0.00 | 0.92±0.17 | 0.02±0.02 | 0.93±0.17 | 0.02±0.01 | 0.95±0.44 | 0.03±0.02 |
| | 0.30 | 0.96±0.97 | 0.02±0.00 | 0.88±0.10 | 0.05±0.01 | 0.91±0.13 | 0.02±0.01 | 0.94±0.39 | 0.02±0.01 |
| | 0.40 | 0.96±0.92 | 0.02±0.01 | 0.77±0.05 | 0.10±0.01 | 0.89±0.10 | 0.04±0.01 | 0.94±0.34 | 0.03±0.02 |
| | 0.50 | 0.96±0.87 | 0.02±0.01 | 0.56±0.03 | 0.26±0.01 | 0.86±0.09 | 0.03±0.01 | 0.95±0.31 | 0.03±0.02 |
| | 0.60 | 0.96±0.81 | 0.02±0.01 | 0.55±0.01 | 0.22±0.00 | 0.85±0.07 | 0.03±0.01 | 0.94±0.27 | 0.02±0.01 |
| | 0.70 | 0.96±0.62 | 0.02±0.01 | ± | ± | 0.77±0.05 | 0.11±0.01 | 0.95±0.19 | 0.02±0.02 |
| | 0.80 | 0.96±0.39 | 0.02±0.01 | ± | ± | 0.71±0.03 | 0.11±0.01 | 0.93±0.12 | 0.03±0.01 |
| | 0.90 | 0.96±0.30 | 0.02±0.01 | ± | ± | 0.76±0.02 | 0.14±0.01 | 0.92±0.08 | 0.03±0.01 |
| | 1.00 | 0.96±0.14 | 0.02±0.01 | ± | ± | 0.63±0.01 | 0.30±0.00 | 0.91±0.04 | 0.05±0.01 |
| Pima indian diabetes | - | 0.75±0.73 | 0.03±0.00 | 0.65±0.49 | 0.06±0.01 | 0.67±0.60 | 0.05±0.01 | 0.71±0.94 | 0.05±0.01 |
| | 0.00 | 0.70±0.98 | 0.04±0.00 | 0.69±0.86 | 0.03±0.01 | 0.71±0.83 | 0.05±0.01 | 0.70±0.99 | 0.05±0.00 |
| | 0.10 | 0.71±0.96 | 0.02±0.01 | 0.70±0.74 | 0.05±0.01 | 0.69±0.79 | 0.05±0.01 | 0.71±0.98 | 0.06±0.00 |
| | 0.20 | 0.72±0.94 | 0.04±0.01 | 0.68±0.64 | 0.04±0.01 | 0.69±0.74 | 0.05±0.01 | 0.71±0.98 | 0.06±0.00 |
| | 0.30 | 0.73±0.86 | 0.02±0.01 | 0.62±0.44 | 0.04±0.02 | 0.68±0.67 | 0.04±0.01 | 0.72±0.96 | 0.02±0.01 |
| | 0.40 | 0.72±0.75 | 0.04±0.01 | 0.55±0.24 | 0.04±0.01 | 0.65±0.55 | 0.03±0.01 | 0.71±0.90 | 0.06±0.01 |
| | 0.50 | 0.73±0.68 | 0.05±0.01 | 0.46±0.16 | 0.06±0.02 | 0.65±0.49 | 0.04±0.01 | 0.70±0.86 | 0.05±0.01 |
| | 0.60 | 0.74±0.60 | 0.04±0.01 | 0.44±0.09 | 0.08±0.01 | 0.63±0.42 | 0.08±0.02 | 0.69±0.78 | 0.04±0.01 |
| | 0.70 | 0.73±0.43 | 0.06±0.02 | 0.44±0.02 | 0.09±0.01 | 0.58±0.29 | 0.05±0.01 | 0.72±0.60 | 0.07±0.01 |
| | 0.80 | 0.72±0.25 | 0.03±0.01 | ± | ± | 0.50±0.17 | 0.05±0.01 | 0.70±0.39 | 0.04±0.01 |
| | 0.90 | 0.73±0.19 | 0.04±0.01 | ± | ± | 0.52±0.12 | 0.08±0.01 | 0.68±0.28 | 0.05±0.01 |
| | 1.00 | 0.71±0.09 | 0.04±0.01 | ± | ± | 0.51±0.04 | 0.10±0.01 | 0.67±0.13 | 0.07±0.01 |
| Heart disease | - | 0.79±0.80 | 0.07±0.01 | 0.71±0.44 | 0.07±0.02 | 0.75±0.58 | 0.07±0.03 | 0.78±1.00 | 0.06±0.00 |
| | 0.00 | 0.77±0.97 | 0.07±0.01 | 0.75±0.78 | 0.08±0.02 | 0.75±0.80 | 0.10±0.01 | 0.76±1.00 | 0.05±0.00 |
| | 0.10 | 0.79±0.96 | 0.06±0.01 | 0.72±0.65 | 0.08±0.02 | 0.75±0.73 | 0.06±0.03 | 0.74±1.00 | 0.05±0.00 |
| | 0.20 | 0.77±0.96 | 0.06±0.01 | 0.72±0.54 | 0.06±0.02 | 0.74±0.69 | 0.05±0.02 | 0.76±1.00 | 0.07±0.00 |
| | 0.30 | 0.79±0.89 | 0.07±0.01 | 0.64±0.35 | 0.08±0.02 | 0.72±0.58 | 0.06±0.02 | 0.74±0.99 | 0.07±0.00 |
| | 0.40 | 0.80±0.80 | 0.05±0.01 | 0.56±0.18 | 0.07±0.02 | 0.70±0.49 | 0.07±0.02 | 0.77±0.96 | 0.08±0.01 |
| | 0.50 | 0.79±0.75 | 0.10±0.02 | 0.44±0.12 | 0.11±0.03 | 0.64±0.42 | 0.09±0.02 | 0.75±0.90 | 0.07±0.02 |
| | 0.60 | 0.81±0.66 | 0.04±0.02 | 0.37±0.06 | 0.16±0.01 | 0.65±0.38 | 0.06±0.03 | 0.77±0.84 | 0.06±0.02 |
| | 0.70 | 0.80±0.48 | 0.08±0.01 | ± | ± | 0.65±0.26 | 0.08±0.02 | 0.76±0.65 | 0.04±0.03 |
| | 0.80 | 0.82±0.30 | 0.06±0.01 | ± | ± | 0.59±0.13 | 0.08±0.01 | 0.77±0.43 | 0.07±0.02 |
| | 0.90 | 0.83±0.22 | 0.07±0.02 | ± | ± | 0.54±0.10 | 0.14±0.01 | 0.75±0.32 | 0.10±0.02 |
| | 1.00 | 0.79±0.10 | 0.06±0.01 | ± | ± | 0.53±0.04 | 0.12±0.01 | 0.78±0.15 | 0.08±0.02 |
| Vehicle | - | 0.69±0.72 | 0.05±0.01 | 0.68±0.48 | 0.05±0.01 | 0.68±0.65 | 0.04±0.01 | 0.70±0.97 | 0.05±0.00 |
| | 0.00 | 0.70±0.97 | 0.03±0.00 | 0.69±0.86 | 0.04±0.01 | 0.70±0.89 | 0.03±0.01 | 0.69±0.99 | 0.05±0.00 |
| | 0.10 | 0.69±0.96 | 0.04±0.01 | 0.71±0.76 | 0.04±0.01 | 0.70±0.86 | 0.03±0.01 | 0.70±0.99 | 0.03±0.00 |
| | 0.20 | 0.69±0.93 | 0.04±0.01 | 0.69±0.66 | 0.04±0.01 | 0.68±0.83 | 0.03±0.01 | 0.70±0.99 | 0.02±0.00 |
| | 0.30 | 0.69±0.85 | 0.04±0.01 | 0.64±0.47 | 0.08±0.02 | 0.68±0.75 | 0.05±0.01 | 0.70±0.97 | 0.06±0.00 |
| | 0.40 | 0.67±0.73 | 0.06±0.01 | 0.52±0.28 | 0.07±0.01 | 0.68±0.66 | 0.05±0.01 | 0.69±0.93 | 0.04±0.01 |
| | 0.50 | 0.68±0.65 | 0.05±0.01 | 0.48±0.18 | 0.04±0.01 | 0.68±0.60 | 0.05±0.01 | 0.69±0.88 | 0.03±0.01 |
| | 0.60 | 0.68±0.58 | 0.03±0.01 | 0.39±0.11 | 0.06±0.01 | 0.66±0.52 | 0.07±0.01 | 0.68±0.82 | 0.04±0.01 |
| | 0.70 | 0.68±0.39 | 0.04±0.01 | 0.34±0.02 | 0.03±0.01 | 0.60±0.37 | 0.04±0.01 | 0.68±0.63 | 0.03±0.01 |
| | 0.80 | 0.65±0.24 | 0.04±0.01 | ± | ± | 0.57±0.23 | 0.06±0.01 | 0.63±0.41 | 0.05±0.01 |
| | 0.90 | 0.62±0.17 | 0.05±0.01 | ± | ± | 0.53±0.16 | 0.04±0.01 | 0.65±0.31 | 0.05±0.01 |
| | 1.00 | 0.57±0.08 | 0.04±0.01 | ± | ± | 0.45±0.07 | 0.06±0.01 | 0.60±0.14 | 0.05±0.01 |

**Table 3.** MSE for regression problems and instance selection compression obtained in the experiments. Threshold denoted as "-" represent normal result without bagging ensemble

| Dataset | thr | RMSE | Compress | RMSE | Compress | Dataset | RMSE | Compress | RMSE | Compress |
|---|---|---|---|---|---|---|---|---|---|---|
| - | | ENN | | CNN | | | ENN | | CNN | |
| Steel C14 noise | - | 1.37±0.72 | 0.07±0.00 | 1.27±0.89 | 0.06±0.00 | Steel C14 | 0.37±0.71 | 0.05±0.00 | 0.27±0.97 | 0.05±0.00 |
| | 0 | 1.32±0.86 | 0.04±0.00 | 1.26±0.99 | 0.03±0.00 | | 0.31±0.89 | 0.05±0.00 | 0.27±1.00 | 0.04±0.00 |
| | 0.1 | 1.32±0.85 | 0.03±0.00 | 1.26±0.99 | 0.03±0.00 | | 0.31±0.89 | 0.06±0.00 | 0.27±1.00 | 0.04±0.00 |
| | 0.2 | 1.33±0.82 | 0.07±0.01 | 1.26±0.97 | 0.03±0.00 | | 0.32±0.83 | 0.04±0.00 | 0.27±0.99 | 0.04±0.00 |
| | 0.3 | 1.35±0.77 | 0.05±0.01 | 1.27±0.91 | 0.01±0.00 | | 0.33±0.78 | 0.05±0.00 | 0.27±0.96 | 0.04±0.00 |
| | 0.4 | 1.37±0.71 | 0.06±0.01 | 1.26±0.81 | 0.04±0.00 | | 0.35±0.70 | 0.05±0.01 | 0.28±0.87 | 0.06±0.00 |
| | 0.5 | 1.38±0.63 | 0.05±0.01 | 1.29±0.64 | 0.02±0.01 | | 0.37±0.61 | 0.05±0.01 | 0.29±0.71 | 0.04±0.01 |
| | 0.6 | 1.39±0.54 | 0.02±0.00 | 1.28±0.42 | 0.04±0.00 | | 0.40±0.51 | 0.05±0.01 | 0.31±0.48 | 0.05±0.00 |
| | 0.7 | 1.41±0.44 | 0.02±0.01 | 1.32±0.22 | 0.04±0.00 | | 0.43±0.41 | 0.06±0.01 | 0.38±0.25 | 0.04±0.01 |
| | 0.8 | 1.45±0.23 | 0.05±0.00 | 1.45±0.02 | 0.10±0.00 | | 0.48±0.22 | 0.05±0.00 | 0.78±0.02 | 0.11±0.00 |
| | 0.9 | 1.46±0.15 | 0.06±0.01 | 1.61±0.00 | 0.11±0.00 | | 0.53±0.15 | 0.07±0.01 | 1.03±0.00 | 0.03±0.00 |
| | 1 | 1.46±0.09 | 0.03±0.01 | ± | ± | | 0.55±0.09 | 0.04±0.01 | ± | ± |
| Crime | - | 0.65±0.65 | 0.11±0.02 | 0.60±0.98 | 0.07±0.00 | Concrete | 0.91±0.76 | 0.14±0.01 | 0.92±0.80 | 0.10±0.01 |
| | 0 | 0.62±0.84 | 0.07±0.01 | 0.59±1.00 | 0.08±0.00 | | 0.91±0.90 | 0.04±0.00 | 0.90±0.90 | 0.10±0.00 |
| | 0.1 | 0.62±0.85 | 0.09±0.01 | 0.61±1.00 | 0.07±0.00 | | 0.90±0.89 | 0.14±0.01 | 0.92±0.90 | 0.10±0.01 |
| | 0.2 | 0.62±0.79 | 0.07±0.02 | 0.59±0.99 | 0.07±0.01 | | 0.91±0.85 | 0.04±0.01 | 0.92±0.85 | 0.11±0.01 |
| | 0.3 | 0.64±0.71 | 0.06±0.01 | 0.61±0.96 | 0.07±0.01 | | 0.92±0.80 | 0.15±0.01 | 0.91±0.80 | 0.13±0.01 |
| | 0.4 | 0.66±0.64 | 0.08±0.02 | 0.60±0.88 | 0.06±0.01 | | 0.90±0.74 | 0.11±0.01 | 0.92±0.74 | 0.12±0.01 |
| | 0.5 | 0.66±0.55 | 0.08±0.02 | 0.61±0.73 | 0.07±0.02 | | 0.92±0.64 | 0.14±0.01 | 0.92±0.64 | 0.06±0.01 |
| | 0.6 | 0.65±0.45 | 0.06±0.02 | 0.63±0.48 | 0.08±0.02 | | 0.89±0.55 | 0.17±0.01 | 0.93±0.50 | 0.15±0.01 |
| | 0.7 | 0.68±0.35 | 0.09±0.02 | 0.63±0.24 | 0.05±0.02 | | 0.90±0.44 | 0.12±0.01 | 0.95±0.39 | 0.12±0.01 |
| | 0.8 | 0.73±0.19 | 0.11±0.01 | 1.03±0.02 | 0.07±0.00 | | 0.92±0.23 | 0.21±0.01 | 1.00±0.17 | 0.07±0.01 |
| | 0.9 | 0.76±0.12 | 0.07±0.01 | ± | ± | | 0.96±0.16 | 0.16±0.01 | 0.97±0.10 | 0.08±0.00 |
| | 1 | 0.78±0.08 | 0.06±0.02 | ± | ± | | 0.98±0.10 | 0.17±0.01 | 1.01±0.06 | 0.07±0.00 |