A VIEW INSIDE THE CLASSIFICATION WITH NON-NESTED GENERALIZED EXEMPLARS

Daniela Zaharie, Lavinia Perian and Viorel Negru
Department of Computer Science, West University of Timisoara - Blvd. Vasile Parvan, nr. 4, 300223, Timisoara, Romania

ABSTRACT

This paper analyses the impact on the classification accuracy of three elements of the Non-Nested Generalized Exemplars (NNGE) classifier: the hyperrectangles splitting procedure, the pruning of non-generalized exemplars and the presentation order of training instances. As a consequence of this analysis some NNGE variants are proposed. A statistical analysis reveals that the proposed variants improve the classification ability of the original NNGE.

KEYWORDS

Instance based learning, nearest neighbor, generalized exemplars, hyperrectangle splitting, pruning.

1. INTRODUCTION

Instance based classifiers rely on using directly the examples from the training set as concept models without constructing abstractions. They proved to be effective (Aha et al., 1991) but the absence of an associated compact explicit model limits their “readability. On the other hand, the rule based approaches can provide comprehensible rule sets. The hybrid variants, based on the concept of generalized exemplars, combine the idea of distance based classification with that of best matching rule. The generalized exemplars are sets of instances which can be interpreted as rules and which allow to decrease the model size and to increase the robustness to noise of instance based classifiers. One of the important contributions to hybrid instance based learning is the Nested Generalized Exemplar (NGE) theory (Salzberg, 1991) which uses both simple instances and generalized exemplars represented as axes-parallel hyperrectangles to model the concepts. The EACH (Exemplar-Aided Constructor of Hyperrectangles) algorithm proposed by Salzberg generalizes the exemplars in an incremental way and allows the generation of overlapping and nesting hyperrectangles (used to represent rules with exceptions). The classification of new examples in EACH is based on a generalized distance involving dynamic weights for hyperrectangles and for attributes. The comparative study between EACH and nearest neighbor classifiers conducted in (Wettschereck and Dietterich, 1995) suggested that the not so good performance of EACH is mainly caused by the existence of overlapping/nesting hyperectangles. As a consequence several variants of the NGE approach have been proposed in (Wettschereck and Dietterich, 1995). The most effective one proved to be BNGE (Batch Nested Generalized Exemplars) which avoids the overlapping between hyperrectangles, uses weights for attributes based on the mutual information and processes the examples in the training set in a batch manner (in order to limit the sensitivity of the algorithm to the processing order of training instances). Further investigations (Martin, 1995) on the negative impact of nesting hyperrectangles led to NNGE (Non Nested Generalized Exemplars) which avoids the generalization of a hyperrectangle if it would cover examples of a different class or split existing hyperrectangles if they prove to cover conflicting instances.

Despite the fact that NNGE proved to be competitive and was implemented in the Weka toolkit (Witten and Frank, 2005) there are very few recent results related to this method. This motivated us to analyze the main components of NNGE and to investigate possibilities to improve its behavior with respect to the classification accuracy and the size of the induced model (measured by the number of generated hyperrectangles). Thus we started by analyzing in depth the most specific component of NNGE, i.e. the splitting procedure and proposed some alternatives. Aiming to reduce the model induced by NNGE we also investigated the impact of a heuristic selection of training instances and of a simple pruning strategy. All these variants are presented in Section 2. Section 3 presents comparative results obtained by applying the proposed NNGE variants and other methods based on generalized exemplars to 22 traditional datasets. Section 4 concludes the paper.

2. LEARNING BASED ON GENERALIZED EXEMPLARS

Let us consider a learning process starting from a set of L examples (training instances), $\{E^1,E^2,\ldots,E^L\}$, each one being characterized by the values of n attributes (the attributes can be numerical, nominal or mixed ones) and a class label. The aim of the learning process is to construct a set of generalized exemplars (hyperrectangles), $\{H^1,H^2,\ldots,H^K\}$. A hyperrectangle usually covers a set of examples and each of its dimensions is specified either by a range of values (in the case of numerical attributes) or by an enumeration of values (in the case of nominal attributes). In the particular case when a hyperrectangle covers just one example it is considered to be a non-generalized exemplar. Each hyperrectangle also has a class label and any example covered by the hyperrectangle and having a different class label is considered a conflicting example. In the NNGE algorithm (Martin, 1995), constructing the set of hyperrectangles starting from the training set is an incremental process where for each example E^j the following three steps are successively applied: *classification* (find the hyperrectangle H^k which is closest to E^j), *model adjustment* (split the hyperrectangle H^k, if it covers a conflicting example) and *generalization* (extend H^k in order to cover E^j, but only if the generalized variant does not cover/overlap a conflicting example/hyperrectangle).

NNGE Algorithm

```
For each example E^j in the training set do:
    Find the hyperrectangle H^k which is closest to E^j    /* Classification step */
   IF D(H^k,E^j)=0 THEN
       IF class(E^j)≠class (H^k) THEN Split(H^k,E^j)       /* Adjustment step */
    ELSE H':=Extend(H^k,E^j)                               /* Generalization step */
      IF H' overlaps with conflicting hyperrectangles
         THEN add E^j as a non generalized exemplar
          ELSE H^k:=H'
```

The classification step is based on the computation of the distance $D(E,H)$ between an example $E=(E_1, E_2, \ldots, E_n)$ and a hyperrectangle H as given in Eq. (1).

$$D(E,H)=\sqrt{\sum_{i=1}^{n}\left(w_i\,\frac{d(E_i,H_i)}{E_i^{\max}-E_i^{\min}}\right)^2} \qquad (1)$$

In Eq. (1) $E_i^{\min}$ and $E_i^{\max}$ define the range of values over the training set which correspond to attribute i (in the case of nominal attributes the length of this range is always 1). H_i is the interval [$H_i^{\min}, H_i^{\max}$] if E_i is a numerical attribute and is a list of values if E_i is a nominal attribute. The distance between the attributes values and the corresponding hyperrectangle "side" is computed depending on the attribute type as described in Eq. (2).

$$d_{\text{nom}}(E_i,H_i)=\begin{cases}0, & E_i \text{ belongs to } H_i\\ 1, & \text{otherwise}\end{cases},\quad d_{\text{num}}(E_i,H_i)=\begin{cases}0, & H_i^{\min}\le E_i\le H_i^{\max}\\ H_i^{\min}-E_i, & E_i<H_i^{\min}\\ E_i-H_i^{\max}, & E_i>H_i^{\max}\end{cases} \qquad (2)$$

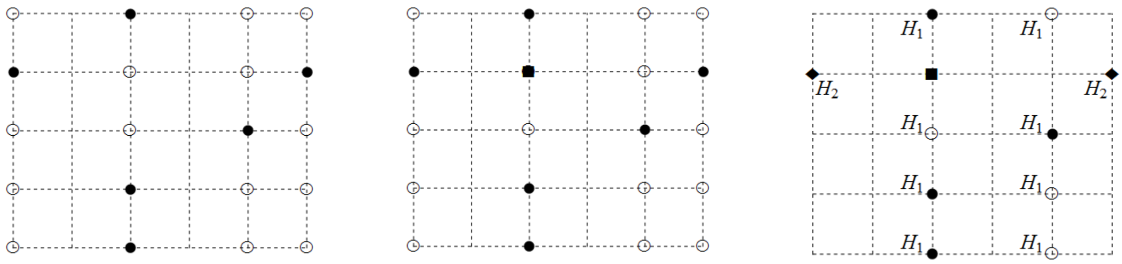
The parameters w_i denote weights corresponding to attributes and can be adjusted during the training process (Salzberg, 1991) or can be set to the mutual information (Wettschereck and Dietterich, 1995). The variant used in this paper is that based on the mutual information.

The adjustment step is applied when an already constructed hyperrectangle covers an example belonging to a different class. In order to avoid the generation of nested hyperrectangles NNGE adjusts the current hyperrectangle such that the conflicting example is excluded. This is realized by splitting the hyperrectangle in a few other hyperrectangles and potentially some isolated instances. This is one of the critical components of NNGE and will be analyzed in more detail in section 2.1.

The generalization step consists of changing the "border" of the closest hyperrectangle having the same class as the training instance in order to cover it. The extension is accepted only if the new hyperrectangle does not overlap with hyperrectangles having a different class. If there is an overlap the training instance is added to the model as a non-generalized exemplar.

2.1 Hyperrectangles Splitting

The splitting process consists of changing one of the dimensions of the hyperrectangle in order to exclude the conflicting example. The choice of the dimension to be changed and the changing approach transform the initial hyperrectangle in a few hyperrectangles and several non generalized exemplars. Since one of the goals is to limit as much as possible the number of hyperrectangles (especially of non generalized ones) the choice of splitting attribute should take this into account. In the case of nominal attributes this is ensured by choosing the attribute for which the value in the conflicting example is less frequent amongst the other examples included in the hyperrectangle. Figure 1 illustrates the case of an initial hiperrectangle (Fig. 1a) containing 6 training examples (filled circles) and covering other 14 instances (empty circles) and of a conflicting example (the black square) covered by the hyperrectangle (Fig. 1b). The value of the first attribute in the conflicting examples is equal to the value of the same attribute in three examples belonging to the initial hyperrectangle. On the other hand the value of the second attribute coincides only with two examples, thus the second atttribute is chosen. This can lead to a pruning of the initial hyperrectangle (H_1 in Fig. 1c) and to the appearance of a new hyperrectangle (H_2 in Fig. 1c) and/or some non-generalized exemplars (if the number of nominal attributes is higher than two).



(a) (b) (c)

Figure 1. Illustration of splitting in the case of nominal attributes. The training examples (all of them belong to the same class) are illustrated by filled circles. (a) The initial hyperrectangle consists of the training examples and their generalizations (empty circles). (b) The example of a different class is represented as a black square. (c) The split is made by using the second attribute: the instances having the same value for the second attribute as the conflicting example define a new hyperrectangle H_2 and the other instances remain in hyperrectangle H_1 (with an adjusted content)

For numerical attributes several selection criteria can be used. In order to describe these criteria let us introduce the following notations: H is the hyperrectangle to be split, E^* is the conflicting example, $T(H)$ is the set of training instances covered by H, A is a subset of $\{1,\ldots,n\}$ and denotes the set of analyzed attributes, i^* denotes the selected splitting attribute. The selection criterion described in Eq (3) corresponds to the variant of NNGE implemented in Weka and consists of choosing the attribute for which the corresponding value of the conflicting attribute is the closest to a margin of the covering hyperrectangle. This is also illustrated in Fig. 2b. In the case of a tie, the attribute leading to the largest number of training examples included in one of the splitting hyperrectangles is chosen (leading to an unbalanced split).

$$i^* = \arg\min_{i\in A} \delta_i, \quad \delta_i = \min\{E_i^* - H_i^{\min}, H_i^{\max} - E_i^*\} \tag{3}$$

The selection criterion described in Eq. (4) corresponds to one of the variants we propose. It chooses the attribute which ensures the most "balanced split", and is also illustrated in Fig. 2b.

$$i^* = \arg\min_{i\in A} r_i, \quad r_i = |(E_i^* - H_i^{\min})/(H_i^{\max} - E_i^*) - 1| \tag{4}$$

The last two variants we propose ("minimal bandwidth" and "maximal bandwidth") are based on analyzing the size of the "free" space between the resulting hyperrectangles as it is expressed by the sum between ε_i^l and ε_i^r defined in Eq. (5). In the case of the "minimal bandwidth" criterion, the attribute *i** is chosen such that $\varepsilon_{i*}^l + \varepsilon_{i*}^r$ is minimal while in the case of "maximal bandwidth" the attribute which maximizes the sum is chosen.

$$\varepsilon_i^l = \min\{E_i^* - E_i;\ E \in T(H), E_i < E_i^*\},\ \varepsilon_i^r = \min\{E_i - E_i^*;\ E \in T(H), E_i > E_i^*\} \quad (5)$$

In all four cases the initial hyperrectangle is divided in at least two hyperrectangles: one for the examples having the value of the splitting attribute strictly higher than the value of the conflicting instance (H_1 in Figs. 2b,2c) and one containing the examples corresponding to strictly smaller values (H_2 in Figs. 2b,2c). The examples having the same value of the splitting attribute as the conflicting instance will either join H_1 or H_2, will form a different hypperrectangle or will remain as non-generalized exemplars (as is illustrated in Figs. 2b, 2c). In more than two dimensions all these four variants could lead to different configurations motivating an analysis of their impact on the resulting classifier behavior.

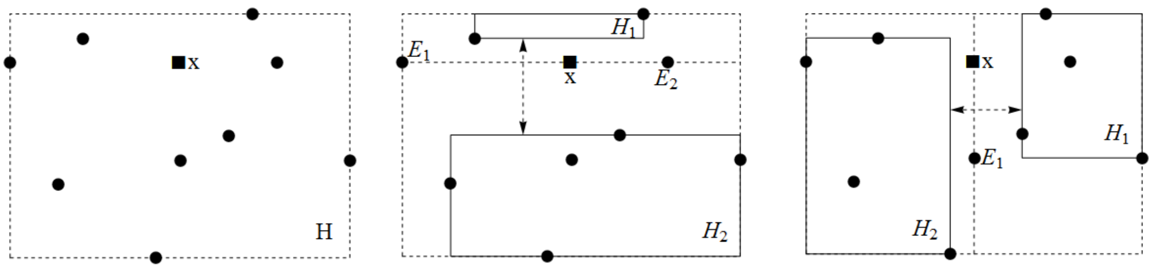


Figure 2. Illustration of splitting variants for numerical attributes. (a) Initially all instances (filled circles) belong to the same hyperrectangle. (b) Effect of splitting by the second attribute (it corresponds to the "closest margin" and to the "maximal bandwidth" strategies) (c) Effect of splitting by the first attribute (it corresponds to the "balanced split" and to the "minimal bandwidth" strategies)

When the attributes are mixed another issue appears: that of choosing between the winning nominal attribute and the winning numerical attribute. In the Weka implementation of NNGE the attribute leading to the highest number of examples included in a hypperrectangle is chosen. Besides this variant we analyzed also a second criterion: minimal volume of resulting hyperrectangles (in order to favor specialized rules and limit the over-generalized hyperrectangles). A comparative analysis of all these variants is presented in Section 3.

2.2 Guiding the Processing Order of Examples

The construction of generalized exemplars is sensitive to the presentation order of the training instances and particularly to the choice of the first instances (the so called seeds). In EACH (Salzberg, 1991) the seeds are chosen randomly from each class without exploiting their relationship with the other instances, while in NNGE (Martin, 1995) the order is arbitrary and there are no seeds (pure incremental variant). Since the aim is to limit the over-generalization caused by large initial hyperrectangles, it seems natural, if the training set is at least partially known, to start by creating compact hyperrectangles in regions with high density of instances belonging to the same class. This idea can be implemented by a simple heuristic strategy consisting of the following components:

- For each instance E_i in the training set construct the largest neighborhood $N(E_i)$ containing only instances having the same class as E_i (it includes all instances E_j satisfying $class(E_i)=class(E_j)$ and $D(E_i, E_j)<D(E_i, E_k)$ with E_k the closest instance belonging to a different class) .
- Sort the training set decreasingly by the size of the corresponding neighborhoods (the size of a neighborhood is given by the number of examples it contains).
- Scan the training list in this order and once an element is processed all its neighbors which were not yet processed are also processed.

This heuristic is based on the same idea as that used in (Garcia, 2009) to construct the initial set of hyperrectangles but here it is used only to sort the instances while the hyperrectangles are constructed following the NNGE procedure. The influence of this strategy on the overall effectiveness and on the size of the induced model is analyzed in Section 3.

2.3 Pruning the Non-Generalized Exemplars

The aim of a pruning process is to obtain a compact model by preserving at the same time as much as possible of the predictive classification accuracy of the initial model. The idea of pruning the non generalized exemplars and the hyperrectangles covering a small number of training instances was analyzed in (Wettschereck, 1994) for BNGE where it was found that removing the non-generalized hyperrectangles has not a significant negative impact on the predictive accuracy.

In the context of NNGE the non-generalized exemplars appear either because their generalization would lead to overlapping or as a result of a splitting process. If they are placed at the borders of the concept models they can help by leading to arbitrarily shaped borders. On the other hand, if they are isolated points surrounded by exemplars corresponding to a different class they are most likely noisy. Therefore it would be useful to analyze the influence on NNGE of the non-generalized exemplars pruning.

3. EXPERIMENTS AND COMPARATIVE RESULTS

In order to analyze the impact of changes on NNGE described in the previous section (different splitting variants, guided processing order of instances and pruning of non-generalized exemplars) experiments on 22 datasets from KEEL repository (Alcalá-Fdez, 2011) were conducted. The datasets are constructed based on those at UCI Machine Learning Repository (http://www.ics.uci.edu/~mlearn/ MLRepository.html) but they are already sliced in folds (e.g. 10) in order to allow a cross-validation based comparison between different methods. The selected datasets and their characteristics are described in Table 1. The experiments were conducted and the statistical analysis was done for all these datasets but, because of limited space, tables with detailed results are presented only for some datasets selected as representative from the point of view of attributes types and size (the datasets with boldfaced name in Table 1).

Table 1. Test data sets (http://www.keel.es/)

Nr.	Name	#Inst.	#Num.	#Nom.	#Cl.	Nr.	Name	#Inst.	#Num.	#Nom.	#Cl.
1	**Australian**	690	8	6	2	12	Hepatitis	80	19	0	2
2	**Breast**	277	0	9	2	13	Iris	150	4	0	3
3	Bupa	345	6	0	2	14	**Led7digit**	500	7	0	10
4	**Cleveland**	297	13	0	5	15	Mammogr.	830	0	5	2
5	**Contracept.**	1473	9	0	3	16	Newthyroid	215	5	0	3
6	**Crx**	653	6	9	2	17	**Pima**	768	8	0	2
7	Dermat.	358	34	0	6	18	Post-oper.	87	0	8	3
8	EColi	336	7	0	8	19	**Sonar**	208	60	0	2
9	Glass	214	9	0	7	20	Wine	178	13	0	3
10	Haberman	306	3	0	2	21	Wisconsin	683	9	0	2
11	Heart	270	13	0	2	22	**Zoo**	101	0	16	7

In order to compare the predictive accuracy of different NNGE variants and of other methods based on generalized exemplars the average accuracy was computed by 10-fold cross validation for all datasets. In order to assess the significance of the difference between the behavior of two methods the Wilcoxon sum rank test was applied as followed. For the investigated methods the standardized differences between the average accuracies ($(A_1 - A_2)/\sqrt{s_1^2 + s_2^2}$) was computed for all data sets. Then the differences were sorted increasingly and the sum of ranks of positive differences (R_+) and the sum of ranks of negative differences (R_-) were computed. Based on the minimum between R_+ and R_- and the critical values of the Wilcoxon test it can be decided if the difference between the methods, suggested by the relationship between R_+ and R_-, is statistically significant or not.

3.1 Influence of the Splitting Strategy

The splitting variants corresponding to numerical attributes described in section 2.1 were tested for all 19 datasets involving numerical attributes. The accuracy values for some selected datasets presented in Table 2 suggest that the splitting variant has an influence on the predictive accuracy of NNGE. In the case of mixed attributes (e.g. Australian, Contraceptive and Crx datasets) the criterion of selection between numerical and nominal attributes seems also to have an influence. In this case the best behavior is ensured by the "minimal bandwidth" variant combined with the criterion of minimal volume of resulted hyperrectangles. Even if there is not a clear winner amongst the splitting variants, the experiments on all datasets involving numerical attributes suggested that the "closest margin" variant which corresponds to the original NNGE implementation (Martin, 1995) leads to smaller accuracy values than the proposed alternatives.

Table 2. Average classification accuracy (%) for different variants of choosing a numerical split attribute (best values are boldfaced, the second best values are italicized)

Data Set		Criteria 1: minimal nr. of isolated points Criteria 2: minimal volume of resulted hyperrectangles				Criteria 1: minimal volume of resulted hyperrectangles Criteria 2: minimal nr. of isolated points			
	Combined	Closest margin	Balanced Split	Max bandwidth	Min bandwidth	Closest margin	Balanced splitting	Max bandwidth	Min bandwidth
Australian	**85.79**	85.07	83.76	82.75	83.04	83.33	83.04	82.89	*85.65*
Cleveland	**57.25**	55.55	56.25	*56.52*	52.86	55.55	56.25	*56.52*	52.86
Contraceptive	44.74	45.14	44.60	45.28	45.08	48.00	46.50	*48.47*	**49.08**
Crx	*84.96*	83.88	83.25	82.48	83.73	83.27	83.53	82.63	**85.45**
Pima	*73.58*	72.79	73.19	71.23	**74.24**	72.79	73.19	71.23	**74.24**
Sonar	64.45	59.64	**68.73**	*65.88*	62.64	59.64	**68.73**	*65.88*	62.64

The results obtained by applying the Wilcoxon test (Table 3) illustrate that the superiority of "balanced split" and "minimal bandwidth" on the "closest margin" is significant while the comparison with the "maximal bandwidth" does not prove that there would be a significant difference.

Table 3. Average accuracy and Wilcoxon test results for the comparison between the "closest margin" splitting variant and the other three splitting variants (19 datasets with numerical attributes)

	Closest margin vs **Balanced split**		Closest margin vs **Max bandwidth**		Closest margin vs **Min bandwidth**	
Average accuracy (%)	76.48	**78.62**	76.48	**77.98**	76.48	**77.88**
Sum of ranks	s=18,R+=36, R-=135		s=18,R+=60, R-=111		s=19,R+=37, R-=153	
Closest margin vs	-		=		-	
p-value	0.015		>0.1		0.009	

Besides the four splitting variants, a combined one is also proposed (accuracy results on second column of Table 2). In this "combined" approach all four variants are tried and that leading to the smallest number of non generalized exemplars is applied. In the following experimental results if the splitting variant is not mentioned this means that the combined approach was used.

3.2 Influence of the guided Processing Order

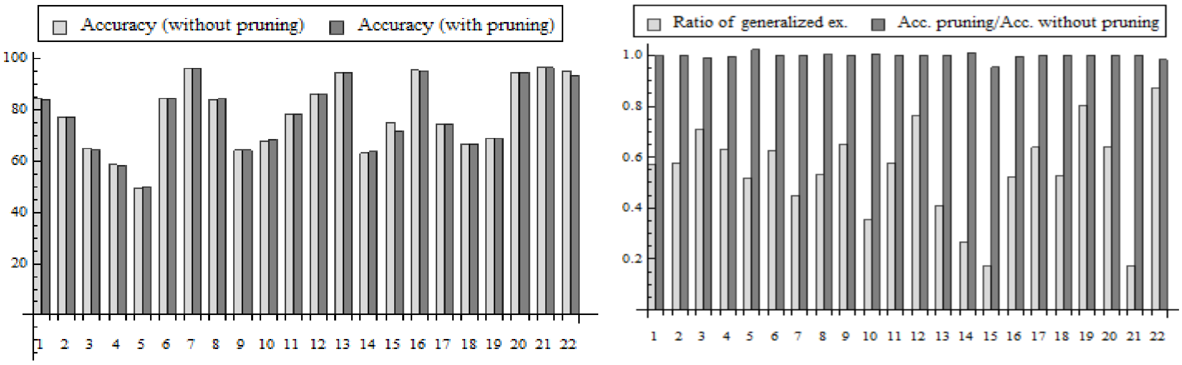
The idea behind the guided processing order described in section 2.2. was to try to limit the occurrence of hyperrectangle splitting steps and consequently to reduce the number of hyperrectangles with a potential positive impact on predictive accuracy. Our expectations were not entirely confirmed by the experimental results. As values in Table 4 show, the only aspect on which the variant with guided processing order is superior to the variant based on an arbitrary processing order is that of the number of resulting hyperrectangles. Even if both the average accuracy and number of splits are better in the case of the guided variant the difference is not statistically significant.

Table 4. Wilcoxon test results for the comparison between the PNNGE (combined splitting and pruned non-generalized exemplars) variant with arbitrary order of examples and the variant based on the guided order (22 datasets)

	Accuracy (%)		Nr. of hyperrectangles		Nr. of splits	
	Arbitrary	Guided	Arbitrary	Guided	Arbitrary	Guided
Average	77.35	**77.95**	48.35	**44.71**	26.88	**26.46**
Sum of ranks	s=20,R+=72, R-=138		s=22,R+=216, R-=37		s=22, R+=94, R-=159	
Comparison	=	=	-	+	=	=
p-value	>0.1		0.001		>0.1	

3.3 Influence of pruning the Non-Generalized Exemplars

Since the ratio of generalized exemplars varies between 0.13 and 0.8 (Fig. 3b) it follows that the pruning of non-generalized exemplars can lead to a significant decrease of the stored exemplars. The largest decrease (around 80% of exemplars are pruned) was identified in the case of large datasets (e.g. 21-Wisconsin and 15-Mammographic datasets) while the smallest one is encountered for smaller datasets (22-Zoo, 19-Sonar , 12-Hepatitis). The difference between the classification accuracy of the non-pruned and pruned variants is not statistically significant (the absolute values are illustrated in Fig. 3a and the ratio of accuracies between the pruned and not pruned variants is plotted in Fig. 3b). In 11 cases out of 22 the accuracy remained unchanged by pruning and in the other cases the Wilcoxon sums of ranks are R+=39, R-=27 with a slight superiority of the un-pruned variant, but without statistically significant difference between them.



(a) (b)

Figure 3. Influence of pruning non-generalized exemplars on the classification accuracy (%) and on the number of final exemplars. The pairs of bars correspond to the datasets described in Table 1

3.4 Comparison with other Methods based on Generalized Exemplars

The proposed variant with combined splitting strategy, arbitrary/guided processing order and pruning of non-generalized exemplars (denoted as PNNGE) was compared with the following methods also based on generalized exemplars: EACH (Salzberg, 1991), INNER (Luaces et al, 2003), RISE (Domingos, 1996), BNGE(Wettschereck, 1994), NNGE (Martin, 1995), EHS (Garcia, 2009). Except for PNNGE for which we use our C++ implementation for the other variants the implementations provided by KEEL and Weka were used. The average accuracies for all these variants are presented for 9 selected datasets in Table 5. In 6 out of 9 cases the proposed PNNGE variants led to higher classification accuracies. The results of the statistical analysis presented in Table 6 show that this superiority is statistically significant in the case of almost all methods (except for the evolutionary approach EHS which on the other hand involves some costly operations and the need to specify the control parameters of the evolutionary process).

Table 5. Comparison between the classification accuracy (%) of the proposed PNNGE versions (combined splitting variant and pruning of non generalized exemplars) and that of other classifiers based on generalized exemplars; best values are boldfaced and the second best values are italicized

Data Set	EACH (KEEL)	INNER (KEEL)	RISE (KEEL)	BNGE (KEEL)	NNGE (Weka)	EHS (Garcia,2009)	PNNGE arbitrary	PNNGE guided
Australian	58.55	**85.79**	80.14	84.34	*84.78*	*84.78*	**85.79**	84.05
Breast	32.37	70.41	69.21	64.17	67.14	*73.46*	*73.46*	**77.19**
Cleveland	-	51.10	49.19	*57.4*	54.20	55.83	57.25	**58.25**
Contraceptive	43.52	46.36	44.94	48.47	-	*49.83*	44.74	**50.10**
Crx	58.23	72.06	81.62	80.48	82.23	*84.64*	**84.96**	84.51
Led7digit	41.19	47.00	*65.20*	65.00	62.8	**68.20**	64.4	63.6
Pima	65.23	72.67	64.18	73.50	70.70	*73.84*	73.58	**74.63**
Sonar	50.33	**76.92**	*76.90*	60.64	68.75	76.50	64.45	68.71
Zoo	84.22	91.30	**96.83**	**96.83**	96.03	93.00	93.5	93.5

Table 6. Wilcoxon test results comparison between the classification accuracy of PNNGE with guided order of examples presentation and the classification accuracy of other classifiers based on generalized exemplars (9 datasets)

	EACH	INNER	RISE	BNGE	NNGE	EHS
s	8	9	9	9	8	9
Sum of ranks (R+)	36	39	38	37	28	23
Sum of ranks (R-)	0	6	7	8	8	22
p-value	0.001	0.027	0.037	0.048	0.097	>0.1
PNNGE(guided) vs	++	+	+	+	+	=

4. CONCLUSION

The analysis of several splitting strategies illustrated that the overall behavior of the NNGE algorithm can be slightly improved especially in the case of datasets containing mixed attributes. Since none of the splitting variants proved to be the best on all test datasets, a combined variant (choosing at each splitting step the strategy leading to the smallest number of non-generalized exemplars) has been proposed and further used to test two other hypotheses: (i) the classification model can be improved by guiding the processing order of training instances; (ii) the model size can be reduced by pruning the non-generalized exemplars, without significantly altering the predictive classification accuracy. Both hypotheses were confirmed by a Wilcoxon sum rank test, in what concerns the size of the induced model. On the other hand, the pruning NNGE (PNNGE) using a combined splitting strategy and a processing order guided by identifying dense homogeneous regions was compared with other six classifiers based on generalized exemplars on nine datasets and proved to be competitive. However the gain in accuracy is not so consistent suggesting that other mechanisms should be also investigated in order to improve the predictive accuracy of classifiers based on non-nested generalized exemplars.

ACKNOWLEDGEMENT

This work was partially supported by the grants POSDRU 21/1.5/G/13798 and PNCDII 12-122/01.10.2008 (ASISTSYS).

REFERENCES

Aha, D.W. et al, 1991. Instance-Based Learning Algorithms. *Machine Learning*, Vol. 6, pp 37-66.

Alcalá-Fdez, J. et al, 2011. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Journal of Multiple-Valued Logic and Soft Computing*, Vol. 17:2-3, pp.255-287.

Domingos, P., 1996. Unifying instance-based and rule-based induction, *Machine Learning*, Vol. 24, 1996 pp. 141- 168.

Garcia, S. et al, 2009. A First Approach to Nearest Hyperrectangle Selection by Evolutionary Algorithms, *Proceedings of 9th International Conference on Intelligent System Design and Applications*, Pisa, Italy, pp.517-522.

Luaces, O. And Bahamonde, A., 2003. Inflating Examples to Obtain Rules. *International Journal of Intelligent Systems*, Vol.18 (11), pp. 1113-1143.

Martin, B., 1995. Instance-Based Learning: Nearest Neighbour with Generalisation**.** *Working Paper Series 95/18* Computer Science, Hamilton,University of Waikato, pp. 90.

Salzberg, S., 1991. A Nearest Hyperrectangle Learning Method. Machine Learning, Vol.6, pp. 251-276.

Wettschereck, D., 1994. A hybrid nearest-neighbor and nearest hyperrectangle algorithm. *Proceedings of European Conference on Machine Learning*, Springer Verlag NY, eds. F. Bergadano, L. De Raedt pp. 323-335.

Wettschereck, D. and Dietterich, T.G., 1995. An Experimental Comparison of the Nearest-Neighbor and Nearest-Hyperrecyangle Algorithms, *Machine Learning*, Vol. 19, pp. 1- 25.

Witten, I.H. and Frank, E., 2005. *Data Mining. Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers.