Fuzzy Rough Set Prototype Selection for Regression

Sarah Vluymans*†, Yvan Saeys†‡, Chris Cornelis*§, Ankur Teredesai¶ and Martine De Cock¶*
*Department of Applied Mathematics, Computer Science and Statistics, Ghent University, Gent, Belgium
Email: {Sarah.Vluymans, Chris.Cornelis, Martine.DeCock}@ugent.be
† VIB Inflammation Research Center, Zwijnaarde, Belgium
Email: {Sarah.Vluymans, Yvan.Saeys}@irc.vib-ugent.be
‡Department of Respiratory Medicine, Ghent University, Gent, Belgium
Email: Yvan.Saeys@ugent.be
§Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain
Email: Chris.Cornelis@decsai.ugr.es
¶Center for Data Science, Institute of Technology, University of Washington Tacoma, Tacoma, USA
Email: {ankurt, mdecock}@uw.edu

***Abstract*—Instance selection methods are a class of preprocessing techniques that have been widely studied in machine learning to remove redundant or noisy instances from a training set. The main focus of such prior efforts has been on the selection of suitable training instances to perform a classification task for crisp class labels. In this paper, we propose a novel instance selection technique termed Fuzzy Rough Set Prototype Selection for Regression (FRPS-R) for solving regression problems, where the outcome is continuous. We use concepts from fuzzy rough set theory and extend the currently well-known fuzzy rough set prototype selection technique to model the quality of all available elements and then use a wrapper approach to select an optimal subset of high-quality instances; thereby generalizing the idea. Our experimental evaluation shows that the application of our proposed instance selection technique can significantly improve the predictive performance of the weighted k-nearest neighbor regression algorithm, in particular when noise is present in the original training set.**

I. Introduction

Instance selection is the task of reducing a dataset of training instances for a machine learning problem to a smaller subset. It is a preprocessing step applied to the data before a specific learning algorithm is used. The k-nearest neighbor classifier (kNN, [1]), for instance, can benefit greatly from the use of instance selection, since it is a lazy algorithm [2] that stores the entire training set as templates instead of learning a classification model. At classification time, an element is assigned to the class to which the majority of its k nearest neighbors belong. This process of nearest neighbors requires an entire pass through the stored training set. In practice it is often approximated using a smaller subset of representative points, forming the motivation for instance selection to prune out the instances that may not contribute to accurate classification. Such instance selection methods specifically designed to be used in conjunction with kNN, are usually referred to as *prototype selection (PS) methods* [3]. Use of PS (a) reduces storage needs for kNN, (b) achieves a speed-up in look-up time and (c) can increase prediction accuracy by removing noisy elements which kNN without preprocessing would give an equal weight in the classification as regular elements.

In this paper, we focus on a PS technique based on fuzzy rough sets that has been shown to improve accuracy of kNN classification. Rough set theory [4] was introduced as a framework for dealing with inconsistencies in information systems, i.e. training instances that have the same values for the input attributes but at the same time have a different outcome. Rough set based prototype selection (e.g. [5]) favors consistent instances over inconsistent ones when it comes to making choices on how to reduce the original training dataset. Fuzzy rough set theory [6] is an extension of rough set theory that allows both input and output attributes to be numerical (continuous) without imposing a need for discretization like traditional rough set theory does. In addition, in fuzzy rough set theory instances can be inconsistent to a degree between 0 and 1 – with 0 being fully consistent and 1 being fully inconsistent. This allows for a more refined preference ranking of instances in the process of prototype selection.

Recently, a fuzzy rough set PS method, called FRPS, was proposed that performs well for kNN classification [7]. Comparison with state-of-the-art prototype selection algorithms suggests FRPS to be the best method w.r.t. prediction accuracy. FRPS has however only been developed for classification problems and is not applicable for regression problems. In this paper, we denote it as FRPS-C. We propose FRPS-R, an extension to FRPS-C that can be used together with the weighted kNN regression algorithm to solve regression problems. In the same way as FRPS-C, our method is a wrapper approach. Based on a newly defined quality measure, fit for regression data, it evaluates a number of candidate prototype subsets. This evaluation is guided by the prediction error of the weighted kNN regression algorithm. We will experimentally show that our method is able to significantly improve the baseline performance of kNN, especially when noise is present in the dataset. Our new method also outperforms a previously proposed preprocessing algorithm for regression.

The remainder of this paper is structured as follows. Firstly, Section II introduces the instance quality measures used in the PS algorithm. In Section III, we recall the FRPS-C algorithm for classification and introduce our proposed extension to regression problems. Section IV presents our experimental study, involving an extensive evaluation of our algorithm as preprocessing method as well as a comparison with a previous proposal. In Section V, we formulate our conclusion and outline future work.

II. Instance quality measures

We consider a decision system $(X, \mathcal{A} \cup \{d\})$ that consists of a universe of instances $X = \{x_1, \ldots, x_n\}$, a set of attributes $\mathcal{A} = \{a_1, \ldots, a_m\}$ and a fixed decision attribute $d \notin \mathcal{A}$. The value of instance x for attribute a is denoted by $a(x)$. Without loss of generality, we assume that all continuous attributes are normalized to values between 0 and 1. We can define two fuzzy similarity relations over X: a similarity relation $R_{\mathcal{A}}$ that assesses the degree of similarity based on the input attributes and a similarity relation R_d that assesses the degree of similarity based on the outcome. For x and y in X, we define

$$R_{\mathcal{A}}(x, y) = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} R_a(x, y), \tag{1}$$

where $R_a(x, y) = 1 - |a(x) - a(y)|$ for a continuous attribute a, and $R_a(x, y) = 1$ if x and y have the same values for a and 0 otherwise for a discrete attribute a [7]. R_d is defined in the same way, i.e. $R_d(x, y) = 1 - |d(x) - d(y)|$ when dealing with a regression task, and $R_d(x, y) = 1$ if x and y have the same values for d and 0 otherwise for a classification task. We assume that all continuous attributes are normalized between 0 and 1 internally, such that we have $R_{\mathcal{A}}(x, y) \in [0, 1]$ and $R_d(x, y) \in [0, 1]$. When d is discrete, i.e. when we are dealing with a classification task, $R_d(x, y) \in \{0, 1\}$.

We now develop a measure to express the predictive ability of an instance. Informally, *an instance y is a good predictor to the extent to which instances that are similar to y w.r.t. the input attributes also have a similar outcome as y*. Let us consider a particular instance z. The more z resembles y w.r.t. the input attributes, i.e. the higher $R_{\mathcal{A}}(z, y)$ is, the higher we expect $R_d(z, y)$ to be. This idea can be expressed using a residual implicator from fuzzy logic, such as the Łukasiewicz implicator $\mathcal{I}$ which is defined as $\mathcal{I}(p, q) = \min(1 - p + q, 1)$ for all p and q in $[0, 1]$. Note that this operator satisfies $\mathcal{I}(0, 0) = \mathcal{I}(0, 1) = \mathcal{I}(1, 1) = 1$ and $\mathcal{I}(1, 0) = 0$, hence, interpreting 0 as *false* and 1 as *true*, it is a generalization of implication from Boolean logic.

Definition 1: **(Degree of agreement)** Let $(X, \mathcal{A} \cup \{d\})$ be a decision system and $R_{\mathcal{A}}$ and R_d the fuzzy similarity relations over X induced by $\mathcal{A}$ and d respectively. Let $\mathcal{I}$ be a residual implicator. For all $y, z \in X$, the formula

$$P_{\mathcal{I}}(z, y) = \mathcal{I}(R_{\mathcal{A}}(z, y), R_d(z, y)) \tag{2}$$

expresses the extent to which the similarity of z and y w.r.t. the input attributes from $\mathcal{A}$ implies the similarity of their outcomes.

We omit the subscript $\mathcal{I}$ in $P_{\mathcal{I}}(z, y)$ whenever the choice of a particular residual implicator $\mathcal{I}$ is clear from the context or not relevant to the discussion. In this work, as in [7], we use the Łukasiewicz implicator recalled above. Note that, because of a well-known property of residual implicators, $P(z, y) = 1$ iff $R_{\mathcal{A}}(z, y) \leq R_d(z, y)$.

Example 1: Consider instances y, z_1, z_2, z_3 and z_4 such that

- $R_{\mathcal{A}}(z_1, y) = 0.1$ and $R_d(z_1, y) = 0.0$
- $R_{\mathcal{A}}(z_2, y) = 0.1$ and $R_d(z_2, y) = 0.9$
- $R_{\mathcal{A}}(z_3, y) = 0.9$ and $R_d(z_3, y) = 0.2$
- $R_{\mathcal{A}}(z_4, y) = 0.9$ and $R_d(z_4, y) = 0.8$

then $P(z_1, y) = 0.9$, $P(z_2, y) = 1.0$, $P(z_3, y) = 0.3$ and $P(z_4, y) = 0.9$. The instances z_1 and z_2 are far away from y in the input space which causes them to have a high value of P, no matter how (dis)similar their outcome is compared with that of y. Elements z_3 and z_4 are close to y in the input space, but z_3's outcome is dissimilar compared with that of y. This raises a red flag w.r.t. the predictive quality of y as a training instance, as reflected in the low $P(z_3, y)$ score, and similarly w.r.t. the predictive quality of z_3, since the definition of $P(z_3, y)$ is symmetrical.

The overall predictive quality $q(y)$ of an instance y can be defined as the minimum of all the $P(z, y)$ scores taken over all instances z in X, such that it intuitively denotes the degree to which y agrees with all other training instances. This quality value is given by:

$$\begin{aligned} q(y) &= \min_{z \in X} \mathcal{I}(R_{\mathcal{A}}(z, y), R_d(z, y)) && (3) \\ &= \min_{z \in X} \min(1 - R_{\mathcal{A}}(z, y) + R_d(z, y), 1) && (4) \end{aligned}$$

This corresponds to the simplified definition of the degree to which y belongs to the so-called fuzzy rough positive region (see [8]). Step (4) is justified by our choice of the Łukasiewicz implicator. If we are solving a classification task, then $R_d(z, y) = 0$ if $d(z) \neq d(y)$ and $R_d(z, y) = 1$ otherwise, hence (4) simplifies to:

$$q(y) = \min_{d(z) \neq d(y)} (1 - R_{\mathcal{A}}(z, y)) \tag{5}$$

i.e. $q(y)$ is the smallest distance between y and any instance z that belongs to a different decision class. In other words, to determine $q(y)$ it is sufficient to go over the ordered list of nearest neighbors z of y w.r.t. decreasing $R_{\mathcal{A}}(z, y)$ and find the first z_0 that belongs to another decision class than y. The quality of y is then $q(y) = 1 - R_{\mathcal{A}}(z_0, y)$. Consequently, the closer y is in the input space to an instance z_0 from another class, i.e. the higher $R_{\mathcal{A}}(z_0, y)$ is, the lower the quality $q(y)$ of y is as a predictor for classification.

Analogously, when we are solving a regression task, then because

$$q(y) \leq \min(1 - R_{\mathcal{A}}(y, y) + R_d(y, y), 1) = 1 \tag{6}$$

and

$$1 - R_{\mathcal{A}}(z, y) + R_d(z, y) \geq 1 \tag{7}$$

whenever $R_{\mathcal{A}}(z, y) \leq R_d(z, y)$, (4) simplifies to:

$$\begin{aligned} q(y) &= \min_{R_{\mathcal{A}}(z,y) > R_d(z,y)} (1 - R_{\mathcal{A}}(z, y) + R_d(z, y)) && (8) \\ &= \min_{R_{\mathcal{A}}(z,y) > R_d(z,y)} (1 - (R_{\mathcal{A}}(z, y) - R_d(z, y))) && (9) \end{aligned}$$

with the implicit assumption that $q(y) = 1$ in case there is no z for which $R_{\mathcal{A}}(z, y) > R_d(z, y)$.

Example 2: To compute $q(y)$ for instance y from Example 1 using (8) we can immediately eliminate z_2 because $R_{\mathcal{A}}(z_2, y) \leq R_d(z_2, y)$. We rank the other instances z in decreasing order of $R_{\mathcal{A}}(z, y) - R_d(z, y)$, i.e.

- $R_{\mathcal{A}}(z_3, y) - R_d(z_3, y) = 0.7$
- $R_{\mathcal{A}}(z_1, y) - R_d(z_1, y) = 0.1$

- $R_{\mathcal{A}}(z_4, y) - R_d(z_4, y) = 0.1$

This makes it clear that

$$q(y) = 1 - (R_{\mathcal{A}}(z_3, y) - R_d(z_3, y)) = 0.3.$$

Equations (5) and (9) show that both for classification and regression the value of $q(y)$ is determined by the decision value of exactly one other instance z in the nearest neighbors of y, due to the use of the minimum operator. In Example 2, this was z_3. In [7] it was experimentally shown for classification tasks that better PS is achieved when the minimum in Equation (3) is replaced by an ordered weighted averaging operator (OWA, [9]).

Definition 2: **(OWA operator)** Assume that a set of observed numerical values $V = \{v_1, \ldots, v_p\}$ needs to be aggregated and that a weight vector $W = \langle w_1, \ldots, w_p \rangle$ is provided for which

- $\sum_{i=1}^{p} w_i = 1$, and
- $w_i \in [0, 1]$ for all $i \in \{1, \ldots, p\}$.

If for all $i \in \{1, \ldots, p\}, c_i$ is the i-th largest value in V, then the OWA_W aggregation of the values in V is given by:

$$\text{OWA}_W(V) = \sum_{i=1}^{p} (w_i \cdot c_i). \tag{10}$$

The OWA-aggregation of the values in V is in other words obtained by first ranking these values in decreasing order and then taking their weighted average.

The aggregated value depends on the choice of weight vector W. In this work, we adopt an approach using inverse additive weights, as it was shown to be an optimal choice for FRPS-C in [10].

Definition 3: **(Inverse additive weight vector)** An inverse additive weight vector of length p is defined as

$$W^p = \left\langle \frac{1}{p \sum_{i=1}^{p} \frac{1}{i}}, \frac{1}{(p-1) \sum_{i=1}^{p} \frac{1}{i}}, \ldots, \frac{1}{1 \sum_{i=1}^{p} \frac{1}{i}} \right\rangle \tag{11}$$

By taking the inverse of the additive values in the denominator, the weights increase, such that smaller to-be-aggregated values get a higher weight, thereby softening the original minimum operator, which assigns weight one to the smallest value and zero-weights to the remaining ones. Using this notion, we redefine the instance quality measures:

Definition 4: **(Quality of an instance for classification)** Let $(X, \mathcal{A} \cup \{d\})$ be a decision system in which d defines a classification task, and let $y \in X$. The predictive quality of y is defined as

$$Q(y) = \text{OWA}_{W^{|B|}}(\{1 - R_{\mathcal{A}}(z, y) \mid z \in B\}) \tag{12}$$

with $B = \{z \mid d(z) \neq d(y)\}$.

Definition 5: **(Quality of an instance for regression)** Let $(X, \mathcal{A} \cup \{d\})$ be a decision system in which d defines a regression task, and let $y \in X$. The predictive quality of y is defined as

$$Q(y) = \text{OWA}_{W^{|B|}}(\{1 - (R_{\mathcal{A}}(z, y) - R_d(z, y)) \mid z \in B\}) \tag{13}$$

with $B = \{z \mid R_{\mathcal{A}}(z, y) > R_d(z, y)\}$.

Example 3: We compute the quality of instance y from Example 1 using Definition 5. In this example $B = \{z_1, z_3, z_4\}$. The values to be aggregated are, ranked in decreasing order, $\langle 0.9, 0.9, 0.3 \rangle$. The inverse additive weights vector of length 3 is $W^3 \approx \langle 0.18, 0.27, 0.55 \rangle$ hence $Q(y) \approx 0.57$. Comparing this to the value $q(y) = 0.3$ obtained in Example 2, the higher value of $Q(y)$ intuitively reflects the behavior for z_1 and z_4 as well, as opposed to solely considering the low value associated with z_3.

When performing the OWA-aggregation, the values to be aggregated are ranked in decreasing order of $1 - R_{\mathcal{A}}(z, y)$ (Definition 4) and in decreasing order of $1 - (R_{\mathcal{A}}(z, y) - R_d(z, y))$ (Definition 5). The values at the tail of these lists get the highest weight when computing the weighted average, i.e. the higher $R_{\mathcal{A}}(z, y)$ (respectively $R_{\mathcal{A}}(z, y) - R_d(z, y)$) is, the more weight it carries.

III. Fuzzy rough based prototype selection for regression (FRPS-R)

In this section, we recall the FRPS-C algorithm from [7] and introduce our proposed extension to regression problems.

A. The FRPS-C algorithm

The aim of FRPS-C is to remove noisy elements from the training dataset such that they will not be used in the classification by kNN and consequently improve the performance of the latter in terms of classification accuracy. In order to do so, it evaluates the quality of all instances by means of the Q value defined in Definition 4. Its aim is to preserve only high-quality instances in the training set. The question arises however which threshold to impose on the quality value of the instances to be included or not. FRPS-C is a wrapper method evaluating each candidate threshold, which is the set of Q values calculated for the training instances. In particular, for each unique Q value, an intermediate set S is constructed consisting of all training elements whose quality exceeds this threshold. Next, the elements in S are used as prototypes in a classification of the full training set by 1NN, by means of leave-on-out validation. We note that this value for k in kNN is fixed by FRPS-C and not adapted to the value used in the posterior classifier. The threshold leading to the set with the highest classification accuracy is used as the final one, meaning that its corresponding set S is the resulting preprocessed training set. As multiple thresholds can lead to the same optimal classification accuracy, the minimal one out of these is selected. This corresponds to a more conservative choice in the removal of noisy instances, as the lowest threshold out of a given set implies the smallest amount of elements to be removed [10].

To sum up, FRPS-C performs the following steps

1) Determine the $Q(x)$ values for all instances $x \in X$, as defined in Definition 4 and store them in G.

2) Remove duplicates from G. All unique values are considered as thresholds τ.
3) For each threshold $\tau \in G$, consider the subset
$$S_\tau = \{x \in X \,|\, Q(x) \geq \tau\}.$$
4) Predict the classes of all training instances with 1NN, using S_τ as prototype set, by means of leave-one-out validation. Determine the corresponding accuracy.
5) Select the thresholds for which this accuracy is maximal.
6) The final prototype set is the one corresponding to the minimal threshold found in the previous step.

B. Extension to regression problems

We extend the FRPS-C algorithm recalled in the previous section for use in regression problems. Our new method, FRPS-R, is used to evaluate the use of fuzzy rough prototype selection on the instance-based local regression method weighted kNN. When estimating the outcome of a target instance x, this method locates its k nearest prototypes in a stored training set and outputs a weighted average of their outcomes. The weights are set to the inverse distance to the target, using a modified Euclidean distance. This measure is taken as the traditional square root of the sum of the squared feature-wise distances, but we ensure that in case of a discrete feature its distance is set to 1 if the values of the elements differ and to 0 otherwise. Concretely,

$$d(x,y) = \sqrt{\sum_{a \in \mathcal{A}} d_a^2(x,y)}, \tag{14}$$

with $d_a(x,y) = a(x) - a(y)$ for a numerical attribute a and

$$d_a(x,y) = \begin{cases} 0 & \text{if } a(x) = a(y) \\ 1 & \text{otherwise,} \end{cases} \tag{15}$$

when a is discrete. We are following [7] in our use of this modified Euclidean distance for kNN. The difference with the similarity measure defined in (1) should not have a negative effect, as kNN is both used internally to evaluate the quality of instance subsets as well as in the final prediction step. The kNN regression method clearly suffers from the same disadvantages as the kNN classifier mentioned above: it has high storage needs, substantial look-up time to locate neighbors and high sensitivity to noise. This makes the preprocessing step a valid candidate to improve its performance.

The FRPS-R algorithm is a straightforward modification of FRPS-C. The Q value from Definition 4 is replaced by the one defined in Definition 5, as we are now dealing with a continuous rather than crisp outcome. Also, instead of internally considering the accuracy of 1NN to determine the optimal threshold on the quality of instances, the algorithm evaluates the MSE of weighted kNN regression, where the value of k is specified by the user. We therefore consider a minimization rather than maximization problem.

Schematically, FRPS-R performs the following steps

1) Determine the $Q(x)$ values for all instances $x \in X$, as defined in Definition 5 and store them in G.
2) Remove duplicates from G. All unique values are considered as thresholds τ.

TABLE I. DATASETS USED IN THE EXPERIMENTAL STUDY. THEIR ATTRIBUTE TYPES (NUMERICAL OR DISCRETE) AND SIZES ARE SPECIFIED.

Dataset	Attr (N/D)	Inst	Dataset	Attr (N/D)	Inst
autoPrice	15 (15/0)	159	friedman	5 (5/0)	1200
anacalt	7 (7/0)	4052	fruitfly	4 (2/2)	125
autoMPG6	5 (5/0)	392	laser	4 (4/0)	993
autoMPG8	7 (7/0)	392	machineCPU	6 (6/0)	209
baseball	16 (16/0)	337	plastic	2 (2/0)	1650
bodyfat	14 (14/0)	252	pollution	15 (15/0)	60
concrete	8 (8/0)	1030	quake	3 (3/0)	2178
cpu	7 (6/1)	209	stock	9 (9/0)	950
dee	6 (6/0)	365	strike	6 (5/1)	625
diabetes	2 (2/0)	43	treasury	15 (15/0)	1049
ele-1	2 (2/0)	495	wankara	9 (9/0)	1609
ele-2	4 (4/0)	1056	wizmir	9 (9/0)	1461
forestFires	12 (12/0)	517			

3) For each threshold $\tau \in G$, consider the subset
$$S_\tau = \{x \in X \,|\, Q(x) \geq \tau\}.$$
4) Estimate the outcomes of all training instances with kNN, using S_τ as prototype set, by means of leave-one-out validation. Determine the corresponding MSE.
5) Select the thresholds for which this MSE is minimal.
6) The final prototype set is the one corresponding to the minimal threshold found in the previous step.

IV. EXPERIMENTAL STUDY

In this section, we evaluate our newly proposed PS algorithm in combination with the weighted kNN regression method. Firstly, we outline the particulars of the experiments, namely the characteristics of the datasets (Section IV-A), the evaluation measure and the details of the statistical analysis (Section IV-B). Section IV-C sets out with a comparison of the performance of weighted kNN with or without preprocessing by FRPS-R. In Section IV-D, we compare the performance of FRPS-R with a previously proposed instance selection method for regression. Lastly, in Section IV-E, as FRPS-C and therefore FRPS-R as well, was introduced as a noise-removal algorithm, we assess its performance in noisy datasets.

A. Datasets

We have selected 25 small to medium sized regression datasets from the KEEL[1] and Weka[2] dataset repositories. They are listed in Table I.

B. Evaluation measure and statistical analysis

We use five fold cross-validation to report the experimental results. We report the MSE attained by the regression method, as well as the reduction in the number of prototypes. The MSE is defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - \hat{f}(x_i))^2,$$

[1] www.KEEL.es
[2] www.cs.waikato.ac.nz/ml/weka/datasets.html

where n is the size of the dataset, $f(x_i)$ is the actual outcome of the instance x_i and $\hat{f}(x_i)$ is the corresponding estimate.

Although rescaled internally to guarantee a correct working of FRPS-R, the actual outcomes of the datasets are drawn from different ranges. In order to analyze all results on the same scale, we therefore use the ratio of the MSEs of the original kNN with kNN after preprocessing. A value greater than 1 implies that the MSE after preprocessing is lower, showing that an improvement in performance was obtained. To verify whether observed differences are significant, we use a Wilcoxon signed-rank test [11]. This is a non-parametric statistical test assessing whether its null hypothesis of an equivalent performance between two methods can be rejected. Intuitively, if the methods exhibit the same behavior, the differences in performance on the datasets should be distributed symmetrically around the median, meaning that one of the methods does not consistently outperform the other. To verify whether this assumption is supported by the observations, a ranking method is used. The absolute values of the differences in performance between the methods are ranked, such that the smallest difference is assigned rank 1 and the largest difference is assigned rank N, where N equals the number of datasets upon which the methods were executed. The sum of the ranks of the positive differences is denoted by $R+$, while $R-$ refers to the sum of the ranks of the negative differences. The sign of the differences is determined by considering one method as 'Method 1' and the other as 'Method 2' and using a 'Method 1 vs. Method 2' setting. The test statistic is defined as the smaller of the two rank sums. The null hypothesis is rejected when the corresponding p-value is lower than a specified significance level α. This value was set to 0.05 in our study.

We perform the statistical test on the obtained ratios. Since we are verifying whether the application of PS significantly improves the MSE of kNN, this is equivalent to testing whether the ratio is significantly larger than 1. In the Wilcoxon test, we therefore compare the vector of ratios (Rat.) to a vector containing a 1 in each position (Ref.).

C. Improvement on kNN

The only internal parameter of FRPS-R is the value of k. It can be set equal to the value used by the regression method (Table II) or, following [7], it can be fixed to 1 (Table III).

The experiments show that FRPS-R is most useful when combined with 1NN. Conducting the 'Rat. vs. Ref.' test for these values, we found $R+ = 256.0$ and $R- = 69.0$. The p-value of this test is 0.015051, from which we can conclude that PS significantly improves 1NN on the 5% significance level.

Both Table II and III show that, for 3NN and 5NN, the application of FRPS-R leads to an average decrease in performance, as the average ratios are smaller than 1. This observation holds for both settings, i.e. both for using the same value of k internally as in the final regression method as when fixing its value to 1. It is clear that the latter leads to a larger decrease in performance, which should come as no surprise, since using the same k value internally and afterward implies that the reduced prototype set is more finely tuned to the specific settings of the regression method.

TABLE II. COMPARISON OF WEIGHTED kNN WITH AND WITHOUT APPLICATION OF FRPS-R, USING THE SAME k VALUE INTERNALLY. RATIO VALUES PRINTED IN BOLD INDICATE THAT THE APPLICATION OF PS RESULTS IN A BETTER REGRESSION PERFORMANCE. THE REDUCTION (RED.) REPRESENTS THE PERCENTAGE OF PROTOTYPES THAT HAS BEEN REMOVED BY PS.

	$k=1$		$k=3$		$k=5$	
Dataset	Rat.	Red. (%)	Rat.	Red. (%)	Rat.	Red. (%)
anacat	**1.0718**	2.20	**1.0290**	1.24	**1.0607**	0.91
autoprice	**1.0006**	3.31	**1.0062**	0.79	0.9570	5.81
autompg6	**1.0297**	6.43	0.9946	1.08	0.9969	0.89
autompg8	**1.0365**	1.40	**1.0049**	0.57	**1.0152**	0.70
baseball	**1.0092**	2.60	0.9667	3.78	0.9646	1.56
bodyfat	0.9728	0.59	1.0000	0.00	1.0000	0.00
concrete	**1.0086**	0.97	0.9973	0.24	0.9971	0.10
cpu	0.4099	4.76	0.6157	4.17	0.5496	4.41
dee	**1.1382**	23.97	**1.0034**	7.19	0.9723	5.96
diabetes	**1.5215**	70.94	**1.0196**	37.88	0.8844	30.24
ele-1	**1.3265**	29.34	0.9228	17.42	0.8950	11.01
ele-2	**1.0419**	5.16	**1.0352**	3.93	0.9997	0.02
forestfires	**1.9894**	16.97	**1.4302**	37.33	**1.2261**	41.14
friedman	0.9878	2.92	1.0000	0.08	1.0000	0.00
fruitfly	**1.9520**	89.20	**1.3551**	68.20	1.2286	68.60
laser	**1.0919**	0.70	0.8975	0.20	0.9435	0.01
machinecpu	**1.0352**	2.52	0.5585	4.55	0.5884	4.31
plastic	**1.1306**	40.91	**1.0220**	5.83	**1.0131**	4.45
pollution	**1.0982**	51.25	0.9683	15.83	**1.0081**	1.25
quake	**1.3221**	99.77	**1.0364**	50.34	**1.0070**	23.86
stock	0.9777	0.71	0.9990	0.21	0.9969	0.08
strike	**1.5638**	2.96	**1.3679**	2.92	**1.2289**	5.68
treasury	0.8492	0.21	0.7712	0.21	0.8597	0.14
wankara	0.9732	0.30	1.0000	0.00	1.0000	0.02
wizmir	0.9964	0.62	0.9962	0.01	1.0000	0.00
Average	**1.1414**	18.43	0.9999	10.57	0.9757	8.45

TABLE III. RESULTS FOR FRPS-R, WHERE k WAS SET TO 3 AND 5 FOR WEIGHTED kNN AND THE INTERNAL VALUE USED IN FRPS-R WAS FIXED TO 1. RATIO VALUES PRINTED IN BOLD INDICATE THAT THE APPLICATION OF PS RESULTS IN A BETTER REGRESSION PERFORMANCE.

	$k=3$	$k=5$		$k=3$	$k=5$
Dataset	Rat.	Rat.	Dataset	Rat.	Rat.
anacat	0.9855	0.9418	friedman	0.9401	0.9003
autompg6	0.9713	0.9565	fruitfly	**1.3923**	**1.2064**
autompg8	**1.0113**	**1.0246**	laser	0.9398	0.8742
autoprice	0.9972	0.9631	machinecpu	0.7682	0.7440
baseball	0.9935	0.9089	plastic	0.9082	0.8343
bodyfat	0.9388	0.9435	pollution	0.7999	0.6634
concrete	**1.0025**	0.9865	quake	**1.0147**	0.9594
cpu	0.6146	0.6784	stock	0.9512	0.9598
dee	0.9482	0.8600	strike	**1.3669**	**1.2290**
diabetes	**1.0252**	0.8748	treasury	0.7712	0.8169
ele-1	0.9905	0.9391	wankara	0.9715	0.9705
ele-2	**1.0049**	0.8716	wizmir	0.9900	0.9814
forestfires	**1.3969**	**1.1937**			
			Average	0.9878	0.9313

With respect to the reduction obtained by the FRPS-R method relative to the size of the original dataset, Table II reports that this is largest when internally using $k = 1$. The average reduction is shown to decrease for higher values of k. We suspect this to be due to the higher susceptibility of kNN to noise for lower values of k. When its value increases, the method is able to handle noise more appropriately and requires a lower removal rate to obtain a good performance.

TABLE IV. MSE OF 1NN, AFTER PREPROCESSING BY FRPS-R ($k = 1$) AND THE MUTINF METHOD. THE RATIO OF THE MSES IS COMPUTED. VALUES PRINTED IN BOLD INDICATE THAT THE APPLICATION OF FRPS-R RESULTS IN THE BEST PERFORMANCE.

Dataset	MutInf	FRPS-R	Rat.
anacat	0.0462	0.0430	**1.0739**
autompg6	10.9871	10.6844	**1.0283**
autompg8	11.7460	10.6530	**1.1026**
autoprice	10001753.7018	10083020.2581	0.9919
baseball	716810.7428	707753.2188	**1.0128**
bodyfat	14.8641	15.3626	0.9676
concrete	101.9250	100.7296	**1.0119**
cpu	2325.1287	5751.1353	0.4043
dee	0.2887	0.2501	**1.1544**
diabetes	0.6503	0.4243	**1.5327**
ele-1	643889.4364	485200.9394	**1.3271**
ele-2	264573.8425	253685.6261	**1.0429**
forestfires	13387.7977	6734.5026	**1.9879**
friedman	6.3537	6.2647	**1.0142**
fruitfly	612.1280	312.2160	**1.9606**
laser	118.9862	108.8274	**1.0933**
machinecpu	4003.8316	3871.0755	**1.0343**
plastic	5.3576	4.7345	**1.1316**
pollution	3203.6280	2730.2616	**1.1734**
quake	0.0654	0.0499	**1.3114**
stock	0.5208	0.5240	0.9939
strike	509801.2256	320915.7376	**1.5886**
treasury	0.0656	0.0758	0.8649
wankara	8.0016	8.1787	0.9783
wizmir	5.9060	5.8994	**1.0011**
Average			**1.1514**

D. Comparison with the state-of-the-art

In the specialized literature, the main focus of instance selection methods has been on classification, not regression. Some embedded methods have been proposed (e.g. [12]) within a regression model, but we propose to compare FRPS-R to the preprocessing method described in [13], as this is a clear, standalone preprocessing method which can be combined with weighted kNN. The method of [14] could be included in our experimental study as well, but a preliminary evaluation showed that it can result in a 100% reduction, rendering it quite unsuitable.

In [13], the authors use a mutual information criterion to select prototypes. Its developers did not propose a custom name for the method, so we will refer to it as MutInf. We have run this method on all datasets, using the parameter values specified by its developers. We have chosen to use 1NN as regression method, as this is the setting most traditionally paired with PS [3].

The average reduction of MutInf is 3.55%. Comparing this to the average reduction of 18.43% by FRPS-R, it is clear that the latter is able to remove considerably more elements. The prediction results of 1NN can be found in Table IV. Inspecting the average ratio, which is defined in this case as the ratio of the MSE after MutInf and the MSE after FRPS-R, it is larger than 1, indicating that MutInf is outperformed by our method. The individual results of the datasets show that this is the case for as many as 19 out of the 25 datasets. We have tested whether the ratio is significantly larger than 1. The Wilcoxon test 'Rat. vs. Ref.' yielded $R+ = 265.0$ and $R- = 60.0$ and its p-value was 0.004604, such that we can indeed conclude that FRPS-R significantly outperforms MutInf.

TABLE V. REDUCTION IN SIZE OF THE TRAINING SET AFTER APPLICATION OF FRPS-R ON NOISY DATASETS.

k	10%	20%	30%	40%	50%
1	0.2439	0.2626	0.3350	0.3716	0.4547
3	0.1716	0.1867	0.2336	0.2560	0.3115
5	0.1518	0.1656	0.2240	0.2109	0.2366

TABLE VI. RATIOS OF MSE ATTAINED BY 1NN ON NOISY DATASETS. VALUES PRINTED IN BOLD INDICATE THAT THE APPLICATION OF FRPS-R RESULTED IN A BETTER REGRESSION PERFORMANCE.

Dataset	10%	20%	30%	40%	50%
anacat	**1.0920**	**1.1877**	**1.0858**	**1.1364**	**1.1127**
autoMPG6	**1.3819**	**1.6147**	**1.5311**	**1.7961**	**1.7177**
autoMPG8	**1.1371**	**1.2090**	**1.3797**	**1.4030**	**1.6409**
autoPrice	**1.0842**	**1.5597**	**1.3422**	**1.2049**	**1.1448**
baseball	**1.0391**	**1.1887**	**1.2617**	**1.5877**	**1.3839**
bodyfat	**1.0767**	**1.0208**	**1.3770**	**1.1515**	**1.1599**
concrete	**1.0504**	**1.1712**	**1.2344**	**1.1461**	**1.2140**
cpu	0.6525	0.5459	0.9555	**1.3019**	**1.1769**
dee	**1.3486**	**1.5033**	**1.5852**	**1.2432**	**1.3635**
diabetes	**1.1935**	**1.1687**	**1.4517**	**1.2950**	**1.5670**
ele-1	**1.4043**	**1.4024**	**1.5109**	**1.5659**	**1.6448**
ele-2	**1.7168**	**1.6820**	**1.6157**	**1.6679**	**1.7024**
forestFires	**2.1592**	**2.1190**	**1.7275**	**1.9519**	**1.8923**
friedman	**1.0998**	**1.3240**	**1.3008**	**1.3116**	**1.3833**
fruitfly	**2.4470**	**2.7689**	**2.3222**	**1.9845**	**2.1052**
laser	**1.2526**	**1.3036**	**1.3700**	**1.2760**	**1.1668**
machineCPU	**1.1729**	0.9726	**1.2169**	0.7747	**1.2172**
plastic	**1.2748**	**1.3751**	**1.2824**	**1.3680**	**1.4353**
pollution	**1.5168**	**1.5357**	**1.1615**	**1.8350**	**1.3279**
quake	**1.7267**	**1.7216**	**1.9079**	**1.7140**	**1.7311**
stock	**1.0744**	**1.3200**	**1.2954**	**1.3459**	**1.5641**
strike	**1.3272**	**1.3039**	**1.1787**	**1.2206**	**1.3314**
treasury	**1.4951**	**1.8511**	**1.6939**	**1.6345**	**1.6303**
wankara	**1.4264**	**1.5244**	**1.5771**	**1.4922**	**1.4929**
wizmir	**1.3844**	**1.5576**	**1.5402**	**1.5821**	**1.5431**
Average	**1.3414**	**1.4373**	**1.4362**	**1.4396**	**1.4660**

E. Noisy data

Finally, we have conducted experiments on datasets contaminated with different levels of noise, in order to evaluate the performance of FRPS-R and weighted kNN in the presence of noise. We consider noise on the outcome attribute, not on the descriptive features. In order to introduce this noise, we select a predefined percentage of original instances and perturb their outcome by adding a number drawn from the standard normal Gaussian distribution and multiplied with the standard deviation of the outcome in the entire dataset. We have opted to construct five noisy datasets for each of the 25 original ones. To this end, we modified the outcome of 10, 20, 30, 40 or 50% of their instances.

We used the version of FRPS-R where the internal k value is set equal to that used in the posterior regression method, as this setting yielded the best results in Section IV-C. We have set k to 1, 3 and 5. Table V presents the average reductions obtained by the preprocessing methods. A similar trend as in Section IV-C can be observed, in that the average reductions lower with increasing values of k. Furthermore, for increasing

TABLE VII. RATIOS OF MSE ATTAINED BY WEIGHTED 3NN ON NOISY DATASETS. VALUES PRINTED IN BOLD INDICATE THAT THE APPLICATION OF FRPS-R RESULTED IN A BETTER REGRESSION PERFORMANCE.

Dataset	10%	20%	30%	40%	50%
anacat	**1.0190**	**1.0706**	**1.0095**	**1.0067**	**1.0001**
autoMPG6	**1.1577**	**1.1323**	**1.2351**	**1.1871**	**1.2370**
autoMPG8	**1.0274**	**1.0761**	**1.0500**	**1.0733**	**1.1423**
autoPrice	**1.0518**	**1.3320**	**1.0615**	**1.0646**	**1.0638**
baseball	0.9931	0.9865	**1.0798**	**1.0863**	**1.0169**
bodyfat	**1.0062**	0.9732	**1.0905**	0.9642	**1.0314**
concrete	**1.0042**	**1.0524**	**1.0188**	**1.0617**	**1.0637**
cpu	0.6828	0.6419	0.8060	0.8339	0.8509
dee	**1.1690**	**1.1002**	**1.2029**	**1.0272**	**1.0461**
diabetes	**1.0010**	0.9202	**1.0877**	**1.0147**	**1.5382**
ele-1	**1.0696**	**1.0719**	**1.1976**	**1.1018**	**1.2464**
ele-2	**1.2529**	**1.2610**	**1.2438**	**1.2553**	**1.2514**
forestFires	**1.2689**	**1.2471**	**1.1908**	**1.2093**	**1.2023**
friedman	**1.0194**	**1.0916**	**1.0536**	**1.1130**	**1.0837**
fruitfly	**1.6250**	**1.7536**	**1.6799**	**1.2943**	**1.5505**
laser	**1.0363**	0.9422	**1.0729**	**1.0760**	**1.0277**
machineCPU	0.6542	0.6754	0.7336	0.7261	0.8640
plastic	**1.1069**	**1.1440**	**1.1120**	**1.1291**	**1.1467**
pollution	0.9786	**1.0323**	0.9443	**1.1867**	**1.1835**
quake	**1.2877**	**1.2288**	**1.3328**	**1.2708**	**1.2701**
stock	**1.0768**	**1.1544**	**1.1720**	**1.1125**	**1.1790**
strike	**1.1802**	**1.1651**	**1.1346**	**1.1736**	**1.1096**
treasury	**1.1864**	**1.2750**	**1.2071**	**1.1997**	**1.2372**
wankara	**1.1444**	**1.2014**	**1.2402**	**1.1914**	**1.1228**
wizmir	**1.1580**	**1.2012**	**1.1838**	**1.2251**	**1.1743**
Average	**1.0863**	**1.1092**	**1.1256**	**1.1034**	**1.1456**

TABLE VIII. RATIOS OF MSE ATTAINED BY 5NN ON NOISY DATASETS. VALUES PRINTED IN BOLD INDICATE THAT THE APPLICATION OF FRPS-R RESULTED IN A BETTER REGRESSION PERFORMANCE.

Dataset	10%	20%	30%	40%	50%
anacat	**1.0154**	**1.0308**	**1.0051**	**1.0024**	0.9935
autoMPG6	**1.0373**	**1.0556**	**1.1831**	**1.1166**	**1.1822**
autoMPG8	**1.0257**	**1.0396**	**1.0349**	**1.0277**	**1.1998**
autoPrice	**1.0203**	**1.3035**	**1.0268**	**1.0599**	**1.0462**
baseball	0.9658	0.9554	**1.0004**	**1.0584**	**1.0096**
bodyfat	0.9677	**1.0551**	**1.0473**	1.0000	0.9883
concrete	**1.0160**	**1.0256**	**1.0058**	**1.0461**	**1.0311**
cpu	0.6900	0.6905	0.7651	0.8636	0.8372
dee	**1.1300**	**1.0426**	**1.0129**	0.9833	0.9780
diabetes	0.7737	0.8607	**1.0047**	0.8613	**1.4468**
ele-1	**1.0336**	0.9663	**1.0914**	**1.0581**	**1.1212**
ele-2	**1.1671**	**1.1549**	**1.1189**	**1.1716**	**1.1142**
forestFires	**1.1284**	**1.1149**	**1.1016**	**1.1177**	**1.1338**
friedman	0.9828	**1.0470**	**1.0301**	**1.0286**	**1.0449**
fruitfly	**1.4367**	**1.5461**	**1.3938**	**1.1517**	**1.3601**
laser	**1.0290**	0.9421	**1.0035**	**1.0101**	**1.0125**
machineCPU	0.6665	0.7418	0.7457	0.7585	0.8232
plastic	**1.0767**	**1.1035**	**1.0797**	**1.0807**	**1.0981**
pollution	0.8088	0.8376	0.9945	0.9455	0.9087
quake	**1.1704**	**1.1358**	**1.2201**	**1.1699**	**1.1753**
stock	**1.0563**	**1.1108**	**1.1366**	**1.0580**	**1.1056**
strike	**1.1376**	**1.1268**	**1.0987**	**1.1176**	**1.0857**
treasury	**1.1207**	**1.1495**	**1.1244**	**1.1303**	**1.1548**
wankara	**1.0950**	**1.0836**	**1.1393**	**1.1000**	**1.0772**
wizmir	**1.0896**	**1.1266**	**1.0994**	**1.1470**	**1.0928**
Average	**1.0256**	**1.0499**	**1.0586**	**1.0426**	**1.0808**

noise levels, the reduction also increases. This implies that the FRPS-R algorithm correctly detects more elements as being noisy.

Tables VI-VIII evaluate the performance of the regression method and present the ratios of the MSEs obtained by weighted 1NN, 3NN and 5NN respectively. Once again, the ratios are taken by dividing the MSE of kNN on the original dataset by that of kNN on the preprocessed dataset. Values larger than 1 imply that the preprocessed dataset resulted in a lower MSE. We observe that, for all classifiers and for all noise levels, preprocessing yields a lower average MSE than no preprocessing, showing the clear use of FRPS-R as a noise removing algorithm for weighted kNN.

Table IX presents the results of the corresponding statistical analysis. Comparing the performance of kNN with and without preprocessing, we are able to conclude that the preprocessing leads to a significant improvement, except for 5NN when only 10 or 20% of noise is present. In all remaining cases, the application of FRPS-R is required for kNN to be able to handle the noise. This set of experiments clearly proves the particular strength of FRPS-R as a noise-removal algorithm.

V. CONCLUSION AND FUTURE WORK

In this paper, we considered the use of PS to improve the performance of the weighted kNN algorithm for regression problems. In the past, PS has been mostly used for classification purposes. We extended the FRPS-C method, a PS technique for kNN classification based on fuzzy rough set theory. This involved a theoretical extension of the fuzzy rough quality value measure used by FRPS-C, as well as the proper modifications to the inner workings of the method in order to handle the continuous rather than crisp outcome.

We experimentally showed that our modified method, denoted as FRPS-R, is able to significantly improve the classification of weighted kNN, especially when noise is present in the dataset. The experiments also allowed us to conclude that the FRPS-R preprocessing method outperforms a previous proposal for instance selection in regression problems with statistical significance.

As future work, we propose to investigate the scalability of this method to large datasets and evaluate its performance both in terms of prediction accuracy and in terms of scalability on very large information systems with millions of instances. Intuitively, this will require modifications in several places. Firstly, the calculation of the quality value is computationally expensive, as the OWA aggregation implies a sort operation of a possibly very large set of values. Secondly, the method relies on the kNN method, that locates nearest neighbors in the entire training set to predict the outcomes. The integration of parallelization procedures in these phases is certainly required. Finally, instead of following FRPS-C and evaluating all candidate thresholds on the quality values, we can reduce this set to a fixed number of well chosen candidates and thereby limit the number of kNN evaluations.

REFERENCES

[1] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[2] D. Aha, "Editorial," in *Lazy Learning*. Springer, 1997, pp. 7–10.

TABLE IX. RESULTS OF THE STATISTICAL ANALYSIS FOR NOISY DATASETS.

k		10%	20%	30%	40%	50%
1	$R+$	311.0	309.0	324.0	319.0	325.0
	$R-$	14.0	16.0	1.0	6.0	0.0
	p	$6.556 \cdot 10^{-6}$	$1.0074 \cdot 10^{-5}$	$1.192 \cdot 10^{-7}$	$8.344 \cdot 10^{-7}$	$5.96 \cdot 10^{-8}$
3	$R+$	267.0	262.0	281.0	283.0	300.0
	$R-$	58.0	63.0	44.0	42.0	25.0
	p	0.00378	0.00613	$8.082 \cdot 10^{-4}$	$6.314 \cdot 10^{-4}$	$5.388 \cdot 10^{-5}$
5	$R+$	216.0	221.0	273.0	229.0	265.0
	$R-$	109.0	104.0	52.0	71.0	60.0
	p	0.15634	0.11994	0.002026	0.02294	0.005581

[3] S. Garcia, J. Derrac, J. Cano, and F. Herrera, "Prototype selection for nearest neighbor classification: Taxonomy and empirical study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 417–435, 2012.

[4] Z. Pawlak, "Rough sets," *International Journal of Computer Information Science*, vol. 11, pp. 341–356, 1982.

[5] Y. Caballero, R. Bello, Y. Salgado, and M. Garcia, "A method to edit training set based on rough sets," *International Journal of Computational Intelligence Research*, vol. 3, no. 3, pp. 219–229, 2007.

[6] D. Dubois and H. Prade, "Rough fuzzy sets and fuzzy rough sets," *International Journal of General Systems*, vol. 17, pp. 191–209, 1990.

[7] N. Verbiest, C. Cornelis, and F. Herrera, "A prototype selection method based on ordered weighted average fuzzy rough set theory," in *Proceedings of the 14th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, 2013, pp. 180–190.

[8] C. Cornelis, R. Jensen, G. H. Martín, and D. Slezak, "Attribute selection with fuzzy decision reducts," *Information Sciences*, vol. 180(2), pp. 209–224, 2010.

[9] R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decisionmaking," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 18, no. 1, pp. 183–190, 1988.

[10] N. Verbiest, "Fuzzy rough and evolutionary approaches to instance selection," Ph.D. dissertation, Ghent University, 2014.

[11] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, pp. 80–83, 1945.

[12] M. Antonelli, P. Ducange, and F. Marcelloni, "Genetic training instance selection in multiobjective evolutionary fuzzy systems: A coevolutionary approach," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 2, pp. 276–290, 2012.

[13] A. Guillen, L. Herrera, G. Rubio, H. Pomares, A. Lendasse, and I. Rojas, "New method for instance or prototype selection using mutual information in time series prediction," *Neurocomputing*, vol. 73, no. 10, pp. 2030–2038, 2010.

[14] I. Rodriguez-Fdez, M. Mucientes, and A. Bugarin, "An instance selection algorithm for regression and its application in variance reduction," in *2013 IEEE International Conference on Fuzzy Systems (FUZZ)*. IEEE, 2013, pp. 1–8.