Selecting Representative Prototypes for Prediction the Oxygen Activity in Electric Arc Furnace

Marcin Blachnik[1], Mirosław Kordos[2], Tadeusz Wieczorek[1], Slawomir Golak[1]

[1] Silesian University of Technology, Department of Management and Informatics, Katowice, Krasinskiego 8, Poland; marcin.blachnik@polsl.pl
[2] University of Bielsko-Biala, Department of Mathematics and Informatics, Bielsko-Biała, Willowa 2, Poland: mkordos@ath.bielsko.pl

Abstract. Selecting a set of representative prototypes in prediction systems enable us to generate prototype based rules (P-Rules), which constitute a very powerful means of providing domain experts with knowledge about the data and the process depicted by the data. P-rules has already proved very useful in classification tasks. This paper investigates application of P-rules to regression problems. The problem of our concern is prediction of oxygen activity in an electric arc furnace during steel scrap melting. For that purpose we use a new algorithm for determining prototype positions, which is based on conditional clustering. Also a comparison between the new algorithm and the classical clustering-based methods for prototype extraction is described.

Keywords: Regression, Nearest Neighbor, Instance Selection, Context Dependent Clustering, Industrial Application, Electric Arc Furnace

1 Introduction

Prototype based rules shortly called P-rules [3] are a concept of representing knowledge extracted from the data as a triple: a set of prototypes, similarity (or distance) measure and appropriate reasoning scheme. Typically there are two different schemes:

- nearest neighbor approach (nearest prototype) - where the system decision is obtained by determining the nearest prototype to the given test instance according to the following rule:

$$\text{If } j = \underset{i=1,\ldots,c}{argmin}\, D(\mathbf{x};\mathbf{p}_i) \text{ Then } C(\mathbf{x}) = C(\mathbf{p}_j) \tag{1}$$

  where $C(\mathbf{x})$ is a function that returns the value of the instance label $\mathbf{x}$, and c is the number of prototypes.
- threshold rules - where each prototype represents an associated respective field defined by a prototype and a threshold value:

$$\text{If } D(\mathbf{x};\mathbf{p}) < \theta \text{ Then } C(\mathbf{x}) = C(\mathbf{p}) \tag{2}$$

  According to that rule if an instance satisfies a given condition (is closer to some prototype then the given threshold) then a particular decision is made. In case when several rules satisfy that condition usually the order of the sequence of rules determine the primary rule.

Although the concept of the nearest neighbor approach is well known for many years [4], we are reusing it in a different way. Many experts not related to data mining were reporting the problem of knowledge representation. For example metallurgical experts face the problem of excessive amount of data, because current furnaces are equipped with tens of sensors sampled every second (what gives 84 thousand of samples per sensor per day). After a year - a typical period of sample acquisition, the number of samples is over 31 mln per sensor! Thus, instance selection is becoming a very important issue.

In P-rule systems the goal is to replace the original subset of instances by as small as possible subset preserving the model accuracy, such that a compromise between the system accuracy and the number of reference instances (prototypes) is reached. The instances selected in this way may be used for further in depth analysis of relationships between the selected objects. For example in metallurgy such prototypes may be used for identification of the time where a given stage of the process is occurring. In that situation mutual position of the prototypes (according to the Voronoi diagrams) can be used to find out what is the current melting stages the process.

Currently most of the applications of P-rules and in general instance selection methods were dedicated to classification problems for example [2, 7, 12]. However, in this paper we show how to approach regression problems and how to determine prototype position efficiently. Typical approaches in this field are based on clustering, where a set of similar instances is replaced by the center of the cluster and is associated with its label obtained by averaging output values of all samples within that cluster. However, this approach is not appropriate for regression problems, because it causes an unnecessary increase of the number of prototypes and it ignores output information during the training. Such problem appears because prototypes are obtained independently of the output labels. To address this issue we propose another solution, which is based on conditional clustering where information about output values is used in the clustering process as a clustering condition.

Our approach has been applied to a dataset describing the metallurgical process, where the task was to predict the amount of active oxygen in the electric arc furnace during the steel scrap meltdown.

The paper is organized as follow: the next section describes the dataset used in our experiments. Section (3) explains how to use the conditional clustering to determine prototype positions. Section (4) presents the obtained results of the application of the proposed method to predict the amount of oxygen activity. The last section (5) concludes the paper, discusses the obtained results and indicates the further research directions.

2 The problem definition and dataset description

The electric arc furnace (EAF) is the main part of the metal scrap recycling process in steel mini-mill plants. EAF uses the heat generated by electric arcs generated between the graphite electrodes and metal scraps to melt the steel. The process of melting scrap metal in EAFs is used for production of high quality steel. The process is performed in several stages. At least four conditions needs to be satisfied before melted steel can be tapped out from the EAF: no solid metal pieces should be present inside the furnace,

the carbon content should be at a required level, the temperature should be at a specified level and required amount of oxygen should be dissolved in the bath. Reactions taking place inside the furnace are much more predictable if the oxygen activity is controlled. Knowledge of the bath oxygen activity during the operation enables operators to predict what reactions are taking place inside the furnace more accurately and to operate furnaces much closer to their optimal operating conditions by minimizing tolerances of the produced steel parameters as well as minimize the furnace worn-out and energy usage. Once the desired steel composition, temperature and oxygen activity are obtained in the furnace, the tap-hole is opened, the furnace is tilted and the steel is poured into a ladle and undergoes the next batch operation (usually a ladle furnace or ladle station). During the tapping process bulk alloy additions are made based on the bath analysis and the desired steel grade. Deoxidizers may be added to the steel to lower the oxygen content prior to further processing.

2.1 Dataset description and preprocessing steps

To predict the value of oxygen activity we had to extract important information from the industrial database. The dataset is populated with measurements obtained from the EAF process controllers and sensors. Every 0.1 second all sensors are sampled and the obtained values are collected in the industrial database. This allows measuring the mass of each steel scrap bucket loaded into the furnace, energy consumed during the process, amount of carbon and oxygen injected into the furnace, temperatures of the bottom of the furnace casing, the time periods where the process gets suspended. All the values are used to construct the dataset.

The output value that should be predicted (oxygen activity) is recorded only few times during the melting process, because it requires turning off the arc and is very expensive. The solution proposed in this paper selects the nearest past values of all sensors for each oxygen measurement, except the buckets weight values which were constant for each individual melt. Then for that melt, instead of directly predicting the absolute value of oxygen activity the derivative is calculated.

The final dataset used in our experiment includes 6263 samples described by 31 attributes that are: derivatives of six sensors, which measure the temperature at the bottom of the furnace casing, derivative of the amount of carbon, oxygen injected to the furnace by five lances and the burners, electric energy consumed by the process and the total weight of the steel scrap.

Because the values gathered from the sensors include many outliers, each attribute after transformation from the row signal into the derivative is transformed with the hyperbolic tangent function to reduce the outlier's influence on the training process [9]. The transformation requires determining the linear coefficient that adjusts the signals to the appropriate range of values. For that propose the Z-transformation in the following form is used:

$$\mathbf{a}_i = tanh\left(\frac{\mathbf{a}_i - \overline{\mathbf{a}}_i}{std(\mathbf{a}_i)}\right) \tag{3}$$

where $tanh()$ - hyperbolic tangent function, $\mathbf{a}_i$ - i-th attribute of the dataset, $\overline{\mathbf{a}}_i$ - mean value of i-th attribute and $std(\mathbf{a}_i)$ - standard deviation of i-th attribute.

3 Instance construction

As mentioned in the introduction we had to extract representative prototypes that can be further used by human experts for in depth investigation. To achieve that goal we are interested in prototypes that can be interpreted independently and to obtain that we use the nearest neighbor decision algorithm. For the same reason we do not use other prototype based algorithms like RBF networks [1] or the family of Reduced Set Suppoort Vector Machines [11], where the predicted output of the system is a linear combination of all prototypes, what reduces the overall comprehensibility of the system.

One of the simplest solution of prototype construction is input data clustering [8]. In the literature many different clustering algorithms are known [5]. One of the most popular clustering methods is based on the optimization of scalar cost function (OSCF), which has very low (linear) computational complexity $O(ncdi)$ where i is the number of iterations (constant), d the dimensionality of the problem (constant), and c is the number of cluster centers, preserving $n \gg c$. An example of this family are: k-means algorithm, expectation maximization algorithm and a family of Fuzzy C-means clustering.

As it was mentioned in the introduction obtaining informative prototypes requires incorporation of system output (($\mathbf{y}$) variable) in the prototype construction process. A typical approach, based on a simple input data clustering does not preserve such information and thus the process needs to be modified. There are several possibilities of the modification. One of them is based on using input data together with output data for the clustering process. In this approach the dataset $\mathbf{Z}$ is defined by concatenation of input and output data $\mathbf{Z} = \mathbf{X} \cup \mathbf{y}$. In this case the output variable $\mathbf{y}$ is considered to be one of the variables of the dataset and as a result the number of attributes of $\mathbf{Z}$ is $m + 1$. Such solution is often used in training the ANFIS (artificial neural fuzzy inference system) [6]. However, because too little attention is paid to the output variable this approach is not sufficient in our case.

Another typical approach is based on discretization of the output $\mathbf{y}$ attribute and then independent clustering of the instances falling into each discretization bean. This approach causes a very strong influence of model output ($\mathbf{y}$) on the obtained prototypes, so in practical applications the obtained prototype position highly depends on the discretization process. To overcame this problem a good solution can be obtained by applying a soft discretization by defining fuzzy membership functions (MF) instead of hard beans and incorporating these soft discretization beans in the clustering process.

A set of clustering methods, which can incorporate such external input in the clustering process can be found in the literature. These methods are often called context clustering methods. For instance Conditional Fuzzy C-means (CFCM) [10], which requires an external variable that should be in the range [0;1] representing the clustering context. In the described approach the clustering process should be repeated for each membership function defined for the output variable and the values of the MF of each input vector should be provided as an external variable. The final cluster centers (prototypes) of each clustering should be concatenated into a single subset of reference points, as described in fig. (1)

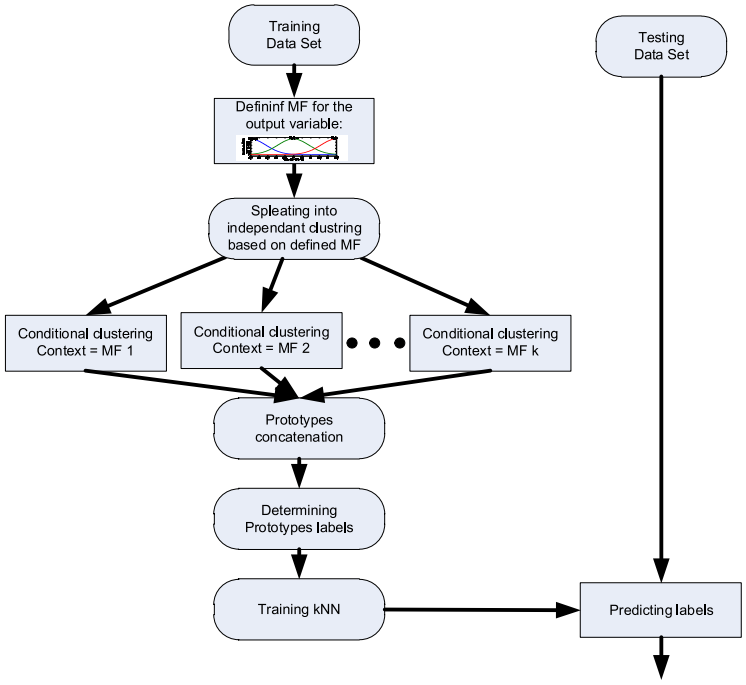


Fig. 1. Diagram of the data analysis process

3.1 Conditional Fuzzy clustering

The most important limitation of FCM in the applications to P-rules system is a lack of information about output variable, which leads to a reduced comprehensibility of the obtained rules. That can be solved by conditional clustering for example by the use of Conditional Fuzzy C-Means. This algorithm allows introducing external knowledge, which represents the influence of certain instance on the clustering process. This knowledge is represented as instance weights $f_k \in [0\,1]$ defined for every instance k in the training set. The value of weight (f_k) is usually calculated as a value of the MF as $f_k = \mu\left(y_k\right)$ of some external variable y_k, where $\mu(\cdot)$ is the function describing the MF.

In CFCM method the cost function remains identical to the FCM one (4),

$$J_m(\mathbf{U},\mathbf{V}) = \sum_{i=1}^{c}\sum_{k=1}^{n}\left(u_{ik}\right)^z \left\|x_k - v_i\right\|_A^2 \tag{4}$$

where $\mathbf{U}$ is a partition matrix of elements u_{ik} representing the value of membership of instance i to the cluster k, matrix $\mathbf{V}$ represents a set of c cluster centers and $z > 1$ is some constant, typically $z = 2$. The partition matrix must satisfy the following requirements:

1^o each vector x_k belongs to the i-th cluster to a degree between 0 and f_k, where $0 \geq f_k \geq 1$:

$$\bigvee_{1 \leq i \leq c} \bigvee_{1 \leq k \leq n} u_{ik} \in [0, f_k] \tag{5}$$

2^o sum of the membership values of k-th vector x_k in all clusters is equal to f_k (in FCM it is equal to 1)

$$\bigvee_{1 \leq k \leq n} \sum_{i=1}^{c} u_{ik} = f_k \tag{6}$$

3^o no clusters are empty.

$$\bigvee_{1 \leq i \leq c} 0 < \sum_{k=1}^{n} u_{ik} < n \tag{7}$$

Cost function (4) is minimized according to $\mathbf{U}, \mathbf{V}$ by the use of Piccard iteration under these conditions by:

$$\bigvee_{1 \leq i \leq c} v_i = \sum_{k=1}^{n} (u_{ik})^z x_k \Bigg/ \sum_{k=1}^{n} (u_{ik})^z \tag{8}$$

$$\bigvee_{\substack{1 \leq i \leq c \\ 1 \leq k \leq n}} u_{ik} = f_k \Bigg/ \sum_{j=1}^{c} \left(\frac{\|x_k - \nu_i\|}{\|x_k - \nu_j\|} \right)^{2/(z-1)} \tag{9}$$

where both equations are iteratively evaluated until the convergence is obtained.

3.2 Defining the clustering context

In the above presented approach defining appropriate clustering context requires defining fuzzy membership functions for the output variable $\mathbf{y}$. This can be done in several ways. One solution is based on manually adjusting fuzzy MF to the output variable. This can be used if the expert can manually define the areas of interests for the output variable. In that case the obtained solution may be very helpful in improving the interpretability of obtained prototypes. Another solution is to adjust the MF automatically. The most naive solution is based on splitting the variable into equal width beans and then setting the position of the MF (for example center of the Gaussian MF) in the middle of each bean with the width of that functions adjusted in this way that the MFs crosses the membership level of 0.5. A more accurate approach can be obtained by the fuzzy clustering of the output variable, which allows for automatically obtaining fuzzy membership values. In that solution FCM clustering is applied for the single variable $\mathbf{y}$. The results of FCM determine centers of the clusters and the partition matrix represents the values of the membership function of each vector to all clusters. In all practical experiments the MF were defined automatically by a simple equal width principle.

3.3 Nearest neighbor rule adaptation

The nearest neighbor prediction rule is simply based on assigning output value identical to the label of the closest prototype. The drawback of that rule is a stair-like output function, where the Voronoi diagram borders are the borders of the plateau of the output function. A simple well known modification of that rule is based on taking into account k nearest neighbors and taking average over all k neighbors. In practise this approach is still not very useful because the predicted values also belong to the finite set of values (output is also a stair-like function), however with much higher number of steps. To overcome this limitation a weighed kNN rule can be used, where the influence of the label of the closest prototypes is inversely proportional to the distance between the k prototypes and the test vector according to the formula (10)

$$c(\mathbf{x}) = \sum_{i=1}^{k} \frac{c\left(\mathbf{p}_i\right) \cdot D\left(\mathbf{x}, \mathbf{p}_i\right)}{\sum_{j=1}^{k} D\left(\mathbf{x}, \mathbf{p}_j\right)} \tag{10}$$

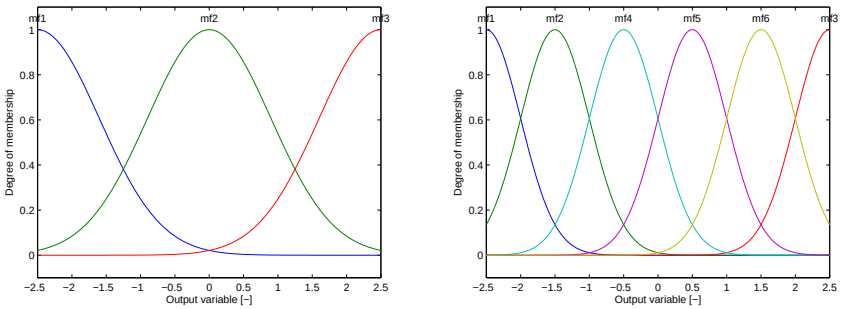
This formula allows pseudo-linear transformation of the k nearest neighbors output values. Unfortunately it reduces the comprehensibility of the system, but on the other hand it increases its accuracy. So the interpretability of the system is inversely proportional to the number of considered nearest neighbors in the decision process (k). In this case a compromise between comprehensibility and accuracy needs to be established. In our experiments we have tested two different solutions; one that preserves the highest comprehensibility $k = 1$ and a second one for $k = 2$ that improves the accuracy, but still preserves simplicity in the decision making process.

4 Numerical examples

In our tests we have compared the behavior of the algorithm described in this paper to the classical clustering methods, to the kNN algorithm and to linear regression as the base rates. In all experiments we have optimized the number of prototypes. The membership functions defined for the oxygen activity (output of the system $\mathbf{y}$) for CFCM clustering are presented in the figure (2). In all the experiments the results of a 10-fold crossvalidation were recorded and a comparison between prototype based methods is presented in figure (3), as well as in the table (1) where also a comparison with other methods like 1NN and kNN is provided. Results marked as W were obtained using weighted version of nearest neighbor rule. For all the methods only the best results are reported in the table. It is important to notice that the obtained values of RMSE are overestimated, because the parameter optimization process was not embedded in the crossvalidation procedure. However, it should not significantly affect the comparison of the methods, because always the same number of parameters were tested for all the algorithms.

5 Conclusions

A new algorithm for optimization of the prototype positions for P-rules was presented. The proposed algorithm uses the knowledge of the output values during the clustering



(a) Three membership functions (b) Six membership functions

Fig. 2. Membership functions defining the clustering context defined for the output variable.

model	RMSE	#P
fcm	0.85±0.038	31
cfcm 3	0.83±0.029	9
W(k=2) cfcm 3	0.84±0.028	9
cfcm 6	0.82±0.032	12
W(k=2) cfcm 6	0.82±0.035	12
kNN	0.80±0.031	k=100
W kNN	0.80±0.031	k=100
kNN	1.07±0.039	k=1

Table 1. Comparison of RMSE and its standard deviation and mean number of prototypes for all compared methods

process. That results in the obtained prototypes being much more representative and in significantly lower error rates. In the presented application we were unable to obtain results as good as for the optimized k-NN classifier. However, we were able to radically reduce the computational complexity, so instead of over 6 thousand vectors, only 12 prototypes were used. The obtained results also prove that determining the correct shape and number of membership functions for the output values is very important. The results show that a higher number of clustering contexts (higher number of membership functions defined for the output variable) leads to much lower error rates, while it does not influence the number of prototypes significantly. In the presented approach the membership functions were defined manually, however in the further research it should be possible to determine them automatically based on the distribution of values of the output variable.

Surprisingly, weighted nearest neighbor rule applied to both: prototypes and to original data did not improve the accuracy, but rather decreased it or increased the variance.

Acknowledgment

The work was sponsored by the Polish Ministry of Science and Higher Education, project No. N N516 442138 and N N508 486638.

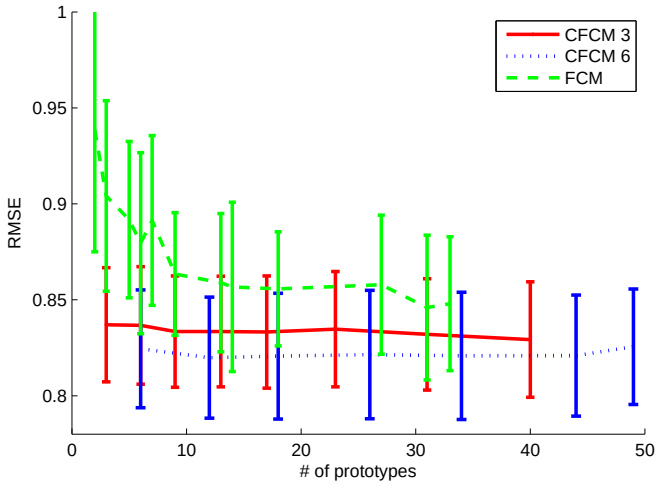


Fig. 3. RMSE as a function of the number of prototypes obtained with FCM and CFCM with 3 and 6 different clustering contexts and the 1NN rule

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)
2. Blachnik, M., Duch, W., Wieczorek, T.: Selection of prototypes rules – context searching via clustering. LNCS 4029, 573–582 (2006)
3. Duch, W., Grudziński, K.: Prototype based rules - new way to understand the data. In: IEEE International Joint Conference on Neural Networks. pp. 1858–1863. IEEE Press, Washington D.C (2001)
4. Duda, R.O., Hart, P.E.: Patter Classification and Scene Analysis. J. Wiley & Sons (1973)
5. Gan, G., Ma, C., Wu, J.: Data Clustering: Theory, Algorithms, and Applications. Series on Statistics and Applied Probability, ASA-SIAM (2007)
6. Jang, J.: Anfis: adaptive-network-based fuzzy inference system. Systems, Man and Cybernetics 23(3), 665 – 685 (1993)
7. Jankowski, N., Grochowski, M.: Comparison of instance selection algorithms. i. algorithms survey. LNCS 3070, 598–603 (2004)
8. L. Kuncheva, J.B.: Nearest prototype classification: Clustering, genetic algorithms or random search? IEEE Transactions on Systems, Man, and Cybernetics C28(1), 160–164 (1998)
9. Kordos M., Blachnik M., Perzyk M., Kozłowski J., Bystrzycki O., Gródek M., Byrdziak A., Motyka Z.: A hybrid system with regression trees in steel-making process. LNCS 6678 (2011)
10. Pedrycz, W.: Conditional fuzzy c-means. Pattern Recognition Letters 17, 625–632 (1996)
11. Schölkopf, B., Smola, A.: Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA (2001)
12. Wilson, D., Martinez, T.: Reduction techniques for instance-based learning algorithms. ML 38, 257–268 (2000)