ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

Applying Mutual Information for Prototype or Instance Selection in Regression Problems

A. Guillen[1], L.J. Herrera[2], G. Rubio[2], H. Pomares[2], A. Lendasse[3], I. Rojas[2] *

1- University of Jaén - Department of Informatics
Spain

2- University of Granada - Department of Computer Technology and Architecture
Spain

3- Helsinki University of Technology - Information and Computer Science Department
Finland

Abstract. The problem of selecting the patterns to be learned by any model is usually not considered by the time of designing the concrete model but as a preprocessing step. Information theory provides a robust theoretical framework for performing input variable selection thanks to the concept of mutual information. Recently the computation of the mutual information for regression tasks has been proposed so this paper presents a new application of the concept of mutual information not to select the variables but to decide which prototypes should belong to the training data set in regression problems. The proposed methodology consists in deciding if a prototype should belong or not to the training set using as criteria the estimation of the mutual information between the variables. The novelty of the approach is to focus in prototype selection for regression problems instead of classification as the majority of the literature deals only with the last one. Other element that distinguishes this work from others is that it is not proposed as an outlier identificator but as an algorithm that determines the best subset of input vectors by the time of building a model to approximate it. As the experiment section shows, this new method is able to identify a high percentage of the real data set when it is applied to a highly distorted data sets.

1 Introduction

The task of selecting the correct subset of input vectors that are included in a training set when classifying, approximating or predicting an output is a relevant task that, if accomplished correctly, can provide storage and computational savings and improve the accuracy of the results.

Three main approaches have been used in order to optimize the set of inputs that the training algorithm will use: *incremental*, *decremental* and *batch*. The incremental approach starts from an empty set of input vectors and defines the training set including input vectors [1]. The opposite perspective is taken in the *decremental* approach that starts considering all the input vectors available

*This work has been partially supported by the Spanish CICYT Project TIN2007-60587 and Junta Andalucia Project P07-TIC-02768

and, following a prefixed criteria, proceeds to remove the non desired instances [2]. The batch method iterates several times before deleting the instance from the training set, setting a flag on the instances that could be removed in next iterations [3]. Recently, many other approaches have been proposed such as evolutive algorithms [4], boosting-based algorithms [5], and pruning techniques [6].

The majority of the research in prototype selection has been focused in classification problems [4], although few works aimed at solving problems for continuous output. For example [7], presents a method to select the input vectors when calculating the output using the k-NN algorithm, however, this methodology does not permit the selection of the input vectors before designing more complex models such as neural networks. In [8], a genetic algorithm is used to select both the feature and the input subsets, however, it is only suitable for linear regression models.

The work developed in this paper is framed within the decremental approach since it considers the whole data set at the beginning. The criteria to remove the input vectors has been taken from the method used to perform feature selection. The problem of finding the adequate set of variables is quite important by the time of designing models to predict, approximate or classify input data. If the set of input data has redundant or irrelevant data, the training can result in overfitted model with poor generalization capabilities [9, 10]. Furthermore, if the dimensionality is not reduced, some local approximator models suffer the curse of dimensionality, making it impossible to design accurate models.

To tackle the feature selection problem. two main streams have been followed: $filter$ and *wrapper* methods. The filter approach consists in a preprocessing of the input data so the model is built after. The wrapper approach attempts to design the model at the same time that performs the variable selection. The concepts of entropy and mutual information (MI) make the Information theory an interesting framework for filtering approaches.

In regression problems, the input and the output values are real and continuous values so additional techniques have to be used to estimate the probability distribution [11]. Although there exists a variety of algorithms to calculate the mutual information between variables, this paper uses the approach presented in [12] which is based on the k-nearest neighbors.

2 Prototype Selection Based on the Mutual Information

This section firstly describes the mutual information, then, the algorithm to perform the prototype selection is introduced.

2.1 Mutual Information

Given a single-output multiple input function approximation or classification problem, with input variables $X = [x_1, x_2, \ldots, x_n]$ and output variable $Y = y$, the main goal of a modelling problem is to reduce the uncertainty on the dependent variable Y. According to the formulation of Shannon, and in the continuous case, the uncertainty on Y is given by its entropy defined as $H(Y) = -\int \mu_Y(y)\log\mu_Y(y)dy$, considering that the marginal density function $\mu_Y(y)$ can be defined using the joint probability density function $\mu_{X,Y}$ of X and Y as $\mu_Y(y) = \int \mu_{X,Y}(x,y)dx$. Given that we know X, the resulting uncertainty of Y conditioned to known X is given by the conditional entropy, defined by $H(Y|X) = -\int \mu_X(x)\int \mu_Y(y|X = x)\log\mu_Y(y|X = x)dydx$. The joint uncertainty on the $[X, Y]$ pair is given by the joint entropy, defined by $H(X, Y) = -\int \mu_{X,Y}(x,y)\log\mu_{X,Y}(x,y)dxdy$. The mutual information (also called cross-entropy) between X and Y can be defined as the amount of information that the group of variables X provide about Y, and can be expressed as $I(X, Y) = H(Y) - H(Y|X)$. In other words, the mutual information $I(X, Y)$ is the decrease of the uncertainty on Y once we know X. Due to the mutual information and entropy properties, the mutual information can also be defined as $I(X, Y) = H(X) + H(Y) - H(X|Y)$, leading to $I(X, Y) = \int \mu_{X,Y}(x,y)\log\frac{\mu_{X,Y}(x,y)}{\mu_X(x)\mu_Y(y)}dxdy$. Thus, only the estimate of the joint PDF between X and Y is needed to estimate the mutual information between two groups of variables.

2.2 Prototype Selection using Mutual Information

The idea that motivates this paper is: since the MI is able to let us know how much information from the output can be retrieved using the different variables starting from a set of input vectors (prototypes), it would be possible that if a significant prototype is removed from the set of input vectors, the amount of MI that could be retrieved would be decreased. On the other hand, if an insignificant prototype is deleted from the original set, the amount of MI should not be decreased. These two sentences are correct, however, there are situations where they might not be completely true. For example, if there are outliers, they will probably provide a significant amount of MI but they should not be considered. On the other hand, if the output of the system remains constant, the amount of information will not fluctuate if similar prototypes are removed.

Thus, in order to make an objective evaluation of how relevant an input vector is, it is necessary to consider the loss of MI relatively to its neighbors in such a way that, if the loss of MI is similar to the prototypes near $\vec{x}_i$, this input vector must be included in the filtered data set. The algorithm proposed to calculate the reduced set of prototypes is described below:
where α_1 is a predefined threshold that determines how different the MI should be respect the neighbors and α_2 is the number of neighbors to be considered in the comparisons.

When calculating how much of MI was lost, two approaches could be taken: 1) to calculate the MI between the complete set of variables and the output or 2) to compute the MI between each variable and the output. With the MI estimator used in the experiments, no difference between those two approaches could be seen, however, other implementations should be analyzed.

Algorithm 1 MI Prototype Selection

```
1. Calculate the $k$ nearest neighbors (k-NN) in the input space of $\vec{x}_i$ for $i = 1...n$
2. for i=1...$d$
        Calculate the mutual information $MIf_i$ when removing $\vec{x}_i$ from $X$
   end
3. Normalize $MIf_i$ in [0,1]
4. for i=1...$n$
        $diff$=0
        for $cont$=1...$\alpha_2$
            $diff$= $|MIf_i| - |MIf_{cont}|$
            if $diff > \alpha_1$
                $Cdiff$=$Cdiff$+1
        end
        if $Cdiff \geq \alpha_2$
            Discard prototype
        else
                Select prototype
        end
end
```

3 Experiment

This section presents the results of applying the new algorithm to a highly distorted data set. The data set was generated syntheticlly so it was possible to know excatly which elements were the originals and which the noisy ones. The target was a one dimensional function (Figure 1 a)) that was generated using a gaussian Radial Basis Function Neural Network (RBFNN) with its parameters randomly chosen.

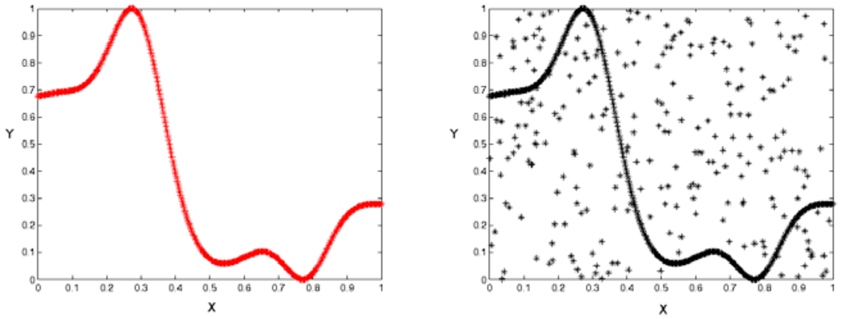

Fig. 1: a) Original target function and b) distorted data set

The original data set consists in 400 prototypes and their corresponding output. This data set was modified adding a set of 250x2 (X, Y) random values in [0,1] from an uniform distribution, obtaining a new data set of 650 prototypes of dimension 1 with one output. This data set is represented in Figure 1 b).

The proposed method was applied using the value 0.05 for the threshold α and 1 for α_2, obtaining a filtered data set of size 447. From the 204 elements removed, 195 were added prototypes and 9 were original prototypes. Thus, the algorithm discriminated the 78.2% of the incorrect prototypes and identified the 97.7% of the original prototypes. Figure 2 depicts the results, where the circles represent the original prototypes and the stars represent the prototypes selected from the distorted data set. If a star is included in a circle, it means that the original prototype was chosen correctly.

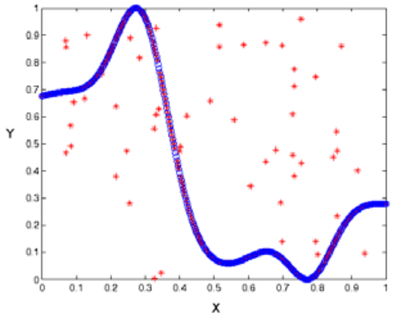

Fig. 2: Filtered data (stars) and original data (circles)

To evaluate the utility a effectiveness of the proposed approach, several RBFNNs were designed using the three different data sets: original, distorted and filtered. The methodology to design the RBFNN was: first, initialize the centers with the algorithm proposed in [13], then apply k-NN to get a first value for the radii and then, apply a local search to make a fine tuning of these parameters. As it was expected, thanks to the prototype selection, the approximation errors (Table 1) that can be obtained are much smaller than if no prototype selection was made.

Data set	error (NRMSE)
original	0.0274
distorted	0.7734
selected	0.4516

Table 1: Approximation errors (Normalized Root Mean Squared Error) obtained when training the networks using the different data sets.

4 Conclusions and Further Work

This paper has presented a possible approach to solve the problem of selecting adequate inputs before using any model to approximate a function. This new method is based on the concept of MI which was used before for feature selection. The main difference between the already existing approaches and the proposed one is that is oriented to data sets with a continuous output value instead of a predefined set of labels and with the global perspective that the MI provides of the complete data set. As the experiment has shown, the method seems quite effective selecting the correct prototypes with a high accuracy. Further work could be done regarding the influence of the two parameters the algorithm has, how to estimate their values building models to evaluate the quality of the selection, and also a comparison among the different ways of calculating the mutual information.

References

[1] David W. Aha. Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms. *Int. J. Man-Mach. Stud.*, 36(2):267–287, 1992.

[2] D. R. Wilson and T. Martinez. Reduction techniques for instance based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.

[3] I. Tomek. An experiment with edited nearest neighbor rule. *IEEE Transactions on Systems, Man and Cybernetics*, 6:448–452, 1976.

[4] H. Ishibuchi, T. Nakashima, and M. Nii. Learning of neural networks with ga-based instance selection. *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, 4:2102–2107 vol.4, 25-28 July 2001.

[5] M. Sebban, R. Nock, and S. Lallich. Stopping criterion for boosting-based data reduction techniques: from binary to multiclass problems. *Journal of Machine Learning Research*, 3:863–865, 2002.

[6] V. B. Zubek and T. G. Dietterich. Pruning improves heuristic search for cost-sensitive learning. In *Proceedings of the International Conference on Machine Learning*, pages 27–34, 2002.

[7] J. Zhang, Y. Yim, and J. Yang. Intelligent selection of instances for prediction functions in lazy learning algorithms. *Artificial Intelligence Review*, 11:175–191, 1997.

[8] J. Tolvi. Genetic algorithms for outlier detection and variable selection in linear regression models. *Soft Computing*, 8(8):527–533, 2004.

[9] E. Liitiäinen, F. Corona, and A. Lendasse. Non-parametric residual variance estimation in supervised learning. In *IWANN 2007*, Lecture Notes in Computer Science. Springer-Verlag, June 20-22 2007.

[10] E. Eirola, E. Liitiäinen, A. Lendasse, F. Corona, and M. Verleysen. Using the delta test for variable selection. In *European Symposium on Artificial Neural Networks, Bruges (Belgium)*, April 2008.

[11] B.V. Bonnlander and A.S. Weigend. Selecting input variables using mutual information and nonparametric density estimation. In *Proc. of the ISANN*, Taiwan, 2004.

[12] L.J. Herrera, H. Pomares, I. Rojas, M. Verleysen, and A. Guillen. Effective Input Variable Selection for Function Approximation. *Lecture Notes in Computer Science*, 4131:41–50, 2006.

[13] A. Guillén, J. González, I. Rojas, H. Pomares, L.J. Herrera, O. Valenzuela, and A. Prieto. Using fuzzy logic to improve a clustering technique for function approximation. *Neurocomputing*, 70(16-18):2853–2860, 2007.