SELECTING CLASSIFIERS TECHNIQUES FOR OUTCOME PREDICTION USING NEURAL NETWORKS APPROACH

Tatiana Shatovskaya

Abstract: This paper presents an analysis of different techniques that is designed to aid a researcher in determining which of the classification techniques would be most appropriate to choose the ridge, robust and linear regression methods for predicting outcomes for specific quasi-stationary process.

Keywords: classification techniques, neural network, composite classifier

ACM Classification Keywords: F.2.1 Numerical Algorithms and Problems

1. Introduction

There are a lot of approaches to building mathematical models for quasi-stationary process with multicollinearity and noisiness. For example, ridge regression is a linear-regression variant that is used for highly correlated independent variables, as is often the case for a set of predictors that are designed to approximate the same function [1]. Ridge regression adds a constraint that the sum of the squares of the regression coefficients be equal to a constant λ. Varying this parameter produces a set of predictors. Robust methods estimation parameters of mathematical model have stability in relation to infringement of requests normality the rests of model. They are insensitive not only to mistakes in a dependent variable, but also take into account a degree of influence of points of factorial space, that is reveal emissions in independent variables that allows to receive effective estimations of the coefficients regression models. For all methods a necessary condition of a solvency of their estimations is symmetry of allocating of mistakes of regression model.

But the main problem for the researcher is how to select an appropriate method for given task. In some cases using only one classification method for choosing the estimation method could not the solve problem. A multitude of techniques exists for modeling process outcomes. But the selection of modeling techniques to use for a given class of process is a nontrivial problem because there are many techniques from which to choose. It could be that the modeling technique used is not the most appropriate for the task and that accuracy can be increased through the use of a more appropriate model. There are many reasons why a model may have low predictive value.

This paper presents an analysis of different techniques that is designed to aid a researcher in determining which of the classification techniques would be most appropriate to choose the ridge, robust and linear regression methods for predicting outcomes for specific quasi-stationary process. We shall try to see that success can be attained with particular architectures on commonly used data for such process.

2. Model Class Combinations

There are many techniques to construct classifiers that will be able to chosen the ridge, robust and linear regression estimation methods. As usual such classifiers build from the same model class, for example using only neural models, decision trees or discriminant function. According to our goal we suggested another approach to building a diverse set of classifiers from different model classes, such as decision trees, nearest neighbor algorithms, linear discriminant function, neural network [2-5]. It is opening question whether classifiers from similar or dissimilar model classes are combined most effectively.

There are many architectures for combination of classifier [2]. One of them is a modular architecture. Modularity is a very important concept in nature. Modularity can be defined as subdivision of a complex object into simpler objects. The subdivision is determined either by the structure or function of the object and its subparts. Modularity can be found everywhere: in living creatures as well as in inanimate objects. Replication and decomposition are the two main concepts for modularity. These concepts are found in concrete objects as well as in thinking. It is often difficult to discriminate sharply between them: replication and decomposition often occur in combination.

Replication is a way of reusing knowledge. Decomposition is often found when dealing with a complex task. It is a sign of intelligent behavior to solve a complex problem by decomposing it into simpler tasks which are easier to manage and then reassemble the solution from the results of the subtasks.

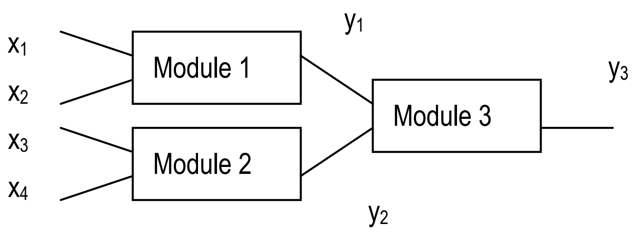

Fig. 1. A Modular Solution

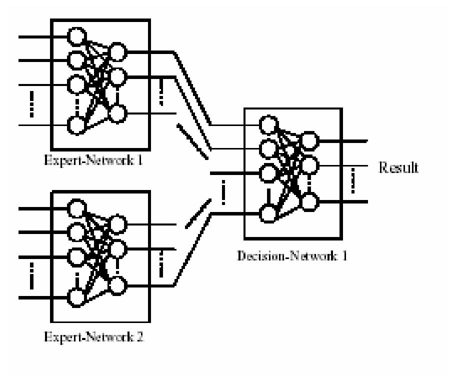

Fig. 2. A Multiple Neural Networks

For example if we choose a neural network as a modular solution we may construct a “building” of neural networks (fig.2).

The term Multiple Neural Networks is used for strongly separated architectures. Each of the networks works independently on its own domain. The single networks are built and trained for their specific task. The final decision is made on the results of the individual networks, often called expert networks or agents. The decision system can be implemented in many different ways: depending on the problem a simple logical majority vote function, another neural network, or a rule based expert system may be employed [6]. The outputs of the expert networks are the input data of the decision network which is trained after the expert networks have been trained. The decision is made according to the outputs of the experts, not directly from the input data. The term Modular Neural Networks (MNN) is very fuzzy. It is used for many different structures [6]. One idea of modular neural network architecture is to build a bigger network by using modules as building blocks. All modules are neural networks. The architecture of a single module is simpler and the sub-networks are smaller than a monolithic network. For this modular approach it is always necessary to have a control system to enable the modules to work together in a useful way. Another idea of modularity is a not-fully connected network. There are many articles and papers published in the field of neural computing [3-6]. An interesting investigation of the relation between structure and function of modular neural networks is given in [7]. The article [7] examines the structural evidence for a modular architecture in the human brain which is given by different psychologists, biologists, and neurologists. Several levels of modularity in the brain are described. Human multitasking abilities and disabilities are explained with the modular and parallel structure of the brain. Individual functions are broken up into sub-processes that can be executed in separate modules without mutual interference. They suggest building more modular artificial neural networks which are similar to the modular structure of the brain [7]. These new architectures may then increase the ability of the network to solve more complex real world problems. Following this motivation for a modular architecture, a new network structure is introduced. The basic building block in this network is the CALM (Categorization and Learning Module) which works on a competitive and unsupervised basis and has the ability to differentiate input patterns in different categories. For a very detailed description of the CALM see [7].

But no meta-generalization scheme is guaranteed to yield neural networks with a minimal generalization error. We concentrates on a recursive modular layered framework for classifier combination or neural networks combination in which the layer of classifiers at each level is used to combine the predictions of the classifiers at the level immediately below.

According to goal of our researching it is suggesting to create two-layer architecture in which the classifiers to be combined are called level-0 classifiers, and the combining classifier is the level-1 classifier. The layering may be iterated to create level-2 classifiers, and so on. Such architecture is a framework for classifier combination in which each layer of classifiers is used to combine the predictions of the classifiers at the immediately preceding layer. A single classifier at the top-most level outputs the ultimate prediction. The classifier at each layer receives as input a vector of predictions of the classifiers in the layer immediately below. While the information passed from layer to layer may take the form of vectors of predictions, confidence values, or other data, we will limit our attention to systems in which only predictions of estimation methods class are passed from layer to layer. We will also limit ourselves to two-layer generalizes, consisting of a set of component classifiers and a single combining classifier that combines the predictions of the component classifiers.

In effect, such combining classifiers are an attempt to minimize generalization error by using the classifiers in higher numbered layers to learn the types of errors made by the classifiers immediately below. The task of the level-1 (and higher) classifiers is to learn to use the contestant predictions to predict more accurately.

Such combining classifiers framework diagram looks like a multilayer neural network diagram (Fig. 1).

There are certainly analogous aspects to the two frameworks. The distinction between them appears to lie partially in the type of information that is passed from the input layer to the succeeding layer and in the granularity of the classifier nodes themselves. In a neural network, an activation value is passed to forward layers, which may or may not be an ultimate prediction or even have some recognizable interpretation. Generally, in the stacked generalization framework, a "full-fledged" class prediction is passed to the combining classifier, and not just a scalar that somehow contributes to a prediction. Also, in other implementations of such classifiers, the classifiers to be stacked are complex, and may be neural networks themselves.

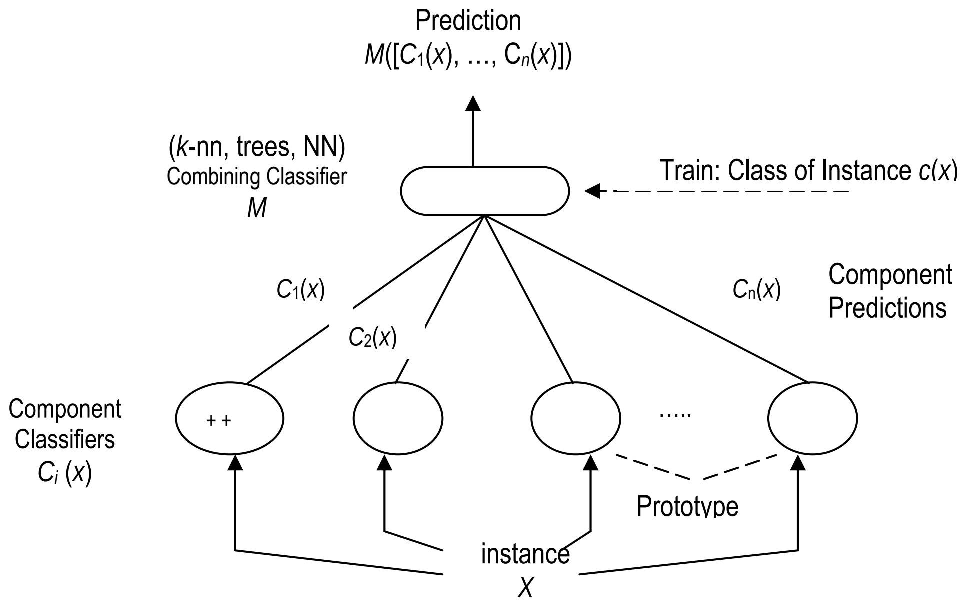

Fig. 3. Classifier Architecture

3. Architecture and Algorithm

We have been given a set of n level-0 (component) learning algorithms, a level-1 learning (combining) algorithm, and a training set of classified instances, T_0. The *n* level-0 learning algorithms should be distinct, so that diverse level-0 classifiers are obtained. Otherwise, no synergy will result from their combination. How to create diverse component classifiers is a fundamental problem for composite classifier construction. Our algorithm has the two phases, training and application.

Training Phase:

1. Train the component classifiers as follows. For each instance in the data set, train each of the *n* level-0 classifiers using the remaining instances. After training, classify the held-out instance using each of the trained level-0 classifiers. Form a vector from the predictions of each of the level-0 classifiers and the actual class of that

instance. These vectors have length *n* + 1, since they have as components the predictions of each of the n level-0 component classifiers and a class label.

2. Train the level-1 classifier, using as the level-1 training set the collection of vectors of the level-0 classifier predictions and the actual classes. This collection has cardinality $|T_0|$, since there is one level-1 training instance corresponding to each level-0 training instance.

3. Since the level-0 classifiers have not been trained on the entire training set, re-train the level-0 classifiers on the entire training set.

Application Phase:

When presented with a new instance whose class is unknown, classify the instance using each of the level-0 classifiers, deriving an input vector for the level-1 classifier. The derived vector is then classified by the level-1 classifier, which outputs a prediction for the new instance. Leave-one-out cross validation is applied in the training phase to ensure that the level-1 algorithm is trained on the generalizations made for unseen data by the level-0 classifiers. Since "generalization" refers to data outside the training set, this observation is memorialized in the name "composite generalization", as opposed to "stacked classification".

In an experiment with combining linear, ridge, robust regression function showed that using 10-fold cross validation to create the level-1 training data yielded slightly more accurate stacked generalizes than when we applied only leave-one-out cross validation. Also in our experiment has been used decision-tree to generate classifiers that make diverse prediction. We combines a set of trees that have been pruned to the *k*-node trees that displayed the smallest training set error, for various choices of *k*. Investigation of the effect of the combination of neural networks with different numbers of units have been performed too. The accuracies of a given model will vary for the different prediction, so have opportunity to compare it on commonly used data.

In our study we used a commonly used data and compare prediction as follow:

- Maximal accuracy prediction: predicted value must lie within a narrow range of actual value.
- Minimal level prediction: actual value is no less than 5 point below predicted value.
- Significant assistance prediction.

Table 1. Accuracy prediction

Model	Accuracy
Combination of Decision trees	55.7%
Combination of Linear discriminant function	68.9%
Combination of Neural network	76.5%
Linear regression	45.8%

The accuracy for each model for the minimal level prediction is higher than those for the same model for the maximal accuracy prediction. Obtained results shows that combined classifier of neural network have the best accuracy prediction. Does this suggest that artificial neural network models should be used for all outcome predictions in class of quasi-stationary process?

For check-up such situation the experiment was designed to test "whether such composite classifier of combination of neural network can be used to separate ridge and robust estimation methods for incomplete input information" using a set of neural network.

As income information from quasi-stationary process with multicolinearity and noisiness for level-0 classifiers used: volume of sample, number of independent variables, degree of multicollinearity, dispersion of a mistake in a dependent variable, ratio of scales of "littering" and basic distributions of the "polluted" distribution of mistakes of model, degree of pollution of independent variables, the form of emissions in independent variables, length of a tail of the "polluted" distribution of independent variables. As a level-0 classifier we used a Probabilistic neural network, Multiple Perceptron Layers, Radial Basis Function for prediction a class or subclass of methods. When an input task is given, the allocator determines which module (neural network) should be used to fulfill this task. Generally, many modules might be selected to fulfill the task together. Each of these selected modules outputs a result based on local computation. The coordinator then gives the final result based on outputs of the modules. If the allocator is so strong that a single module can always be correctly selected to perform a given task, the coordinator can be removed. If, on the other hand, the allocator is so weak that all modules must be used to fulfill a task, a strong coordinator would be useful to make the final judgment. Interesting enough, most existing nets are different from each other simply because their allocators or coordinators are stronger or weaker.

For a level-1 classifier as income information has been used a set of criteria of estimation method accuracy. In the table 4 shows the error rate of prediction the most effective method estimation on every level of classifier.

Table 2. Error rate

Model	Error rate
Probabilistic neural network	0.13 %
Multiple Perceptron Layers	0.15 %
Radial Basis Function	0.2 %

Working within this combined classifier on a difficult incoming data from the quasi-stationary process with multicolinearity and noisiness, composite classifier using a probabilistic classifier and a neural network attained accuracy not achieved by any other learning algorithm or modelling techniques. Does this even suggest that the NN-models should use for all outcome of all quasi-stationary process with multicolinearity and noisiness?

It is necessary to note that in choosing a modeling technique we must weigh the costs of the techniques against the accuracies of the techniques. While it may be cost effective for the minimal level prediction to use an NN-model to gain an additional 5-6% in accuracy, it may not be cost effective to use an NN-model or decision trees model for the maximal assistance prediction. In creating a neural network model there are a large number of decisions that must be made, including: Which learning algorithm should be used? Which architecture? How many layers? Which activation functions? What learning rate? How long to train? And so on. The large number of decisions means that there is a very large space of possible neural networks for a given data set. In creating a neural network model the goal is to find the best network by searching through this large space. On our data set the Probabilistic neural network and Counter propagation neural network has the maximal accuracy, but on the other data set it's not necessary. But the idea to composite such type of classifier or to composite classifier that belongs to different type of model by using the recursive-layered framework allows to minimize the error rate of classification.

Bibliography

1. Лесная Н.С., Репка В.Б., Шатовская Т.Б. Метод выбора эффективных процедур оценивания параметров моделей квазистационарных процессов в нейросетевой экспертной системе // Радиотехника. Всеукраинский межведомственный научно-технический сборник. - Харьков. 2001. - № 119. - С. 195-198.
2. Skalak D.B. Prototype selection for composite nearest neighbour classifiers. Neurological Research 2001; 20. P. 116-328.
3. Breiman L., Friedman J.H., Olshen R.A., Stone C.J. Classification and regression tress. Belmont, CA: Wadsworth. 1984.
4. Lang E.W., Pitts L.H., Damron S.L., Rutledge R. Outcome after severe head injury: Analysis of prediction based upon comparison of neural network versus logistic regression analysis. Neurological Research 1997; 19. P. 274-280.
5. Grisby J., Kooken R., Hershberger J. Simulated neural network to predict outcomes, cost and length of stay among orthopaedic rehabilitation patients. Arch. Phys. Med. Rehabil. 1994. Vol. 75. P. 1077-1082.
6. Manchester Metropolitan University. Department of Computing. Report, September1997. A Modular Neural Network Architecture with Additional Generalization Abilities for High Dimensional Input Vectors.
7. Happle H. and Murre. Design and Evolution of Modular Neural Network Architectures. 2000. Vol. 75. P. 256-277

Authors' Information

Tatiana Shatovskaya - Department of Software Engineering, Kharkiv National University of Radioelectronics, Computer Science Faculty, 61166, Kharkiv, Lenin avenue 14, mywork@kture.kharkov.ua