

Ensemble Learning with Local Experts

Mahdi Milani Fard

Faculty of Electrical and Computer Eng., School of Engineering, University of Tehran

m.milanifard@ece.ut.ac.ir

Abstract

Ensemble learning methods have received considerable attention in the past few years. Various methods for combining several learning experts have been developed and used in different domains of machine learning. Many works have focused on decision fusion of different experts. Some methods try to train all the experts on the same training data and then use statistical techniques to combine the results so that the overall decision is of better accuracy. This paper presents a method in which the experts are not trained on the same data set, but rather they are trained locally with a subset of the training data. Behavioral partitioning is used here as the means to divide the problem space. Different methods are discussed for combining the results. Simple implementation of the method shows results comparable to those of similar methods.

Keywords

Ensemble Learning, Local Learning, Behavioral Partitioning, Decision Fusion

1. Introduction

In the last decade, methods for combining multiple experts (Multi Expert Systems, MES) have been widely studied. Different approaches have focused on different factors of such systems [1]. Among these, ensemble learning methods have received considerable attention and have been employed in different domains of learning ranging from recognition of handwritten numerals [18] to engine knock detection [20].

There have been different approaches to this domain. In most cases, the MES method considers that all the experts are trained on the same training data. So the task would be to combine the results of these experts on a given input [11] [12][13][14][15][16]. This is mostly the case for ensemble classification techniques in which MES is proved to provide more reliable results [1][2][12]. Fusion mechanisms in ensemble classification systems have widely studied and many rigorous statistical techniques have been developed in such areas. The task would be easier if each classifier can provide a

confidence level for its output, in which case it would be easy to calculate the result in the way it minimizes the estimated error [1]. Rejecting the input would also be possible in this case [17].

Regression is a more challenging problem in ensemble methods. Most studies on ensemble regression have focused on hierarchical learning structures, aka Hierarchical Mixture of Experts, HME [4] [5]. In HME methods experts are placed at the leaves of a tree structure and internal nodes are combiners. Both experts and combiners use simple learning models such as Generalized Linear Model, mostly referred to as GLIM. GLIM is simple linear model with a single nonlinearity at the output:

$$y_i = f(U_i x) \quad (1)$$

Where U_i is a weight matrix and f is a fixed continuous nonlinearity [4].

The experts in HME are set with simple models because the overall model of the structure becomes extremely complex as the size grows. Even with such simple models it's not possible to train all the experts and combiners at the same time. With such complex structure, the simple gradient descent method fails to converge to a good result or may take a long time for that. Thus iterative optimization techniques such as EM algorithm are used to train HME models [5] [6].

Unlike most ensemble classification methods, HME structures try to train the experts along with the combiners. This makes more sense when we consider the fact that simple learning models used in HME can provide neither enough confidence information nor much statistical data to be used in the combiners and thus it will be a difficult task to combine the results of pre-trained regression systems.

This paper proposes a method in which the experts can be trained prior to the combiners. Behavioral partitioning is the main method used for this model, which shall ease the problems with the training of experts and combiners that was stated above. The work goes further as it can be generalized to use not only HME, but also other fusion structures.

2. Ensemble Learning

The idea of local learning has been around for a long time. Many learning algorithms are based on local learning method [3]. Here there are expert that are only trained on a subset of the problem domain. These experts are then used, along with a combiner, to provide a global solution to the problem. This, can be seen as an example of the general divide and conquer algorithm.

Once the ensemble is trained, the only thing to do is to feed the input into the selection mechanism and choose the appropriate expert for that input. Then we can use the selected expert to produce the result.

2.1. Behavioral Partitioning

There are two important challenges in this domain. First, there is a need to divide the problem space into some subsets. Second, there should be a way to combine the results. These problems are not always easy to solve, as they cannot be done separately in most cases. The parameters for experts and selectors are correlated and cannot be optimized separately. This usually leads to the problem of complex learning models, which makes the training much more difficult than that of single expert systems. Behavioral partitioning is a method that tries to overcome this problem by separating different phases of the training method. The idea is to find an optimum division of the problem space while the experts are being trained. Once the division is done, it would be easy to train the fusion mechanism. The idea of a dynamic division of the problem space is based on a simple algorithm similar to the classic vector quantization technique.

The VQ algorithm is as follows: Given a set of vectors in an n -dimensional space, we are to find a division in the space that would classify the vectors so that close vectors (in terms of Euclidean distance) are classified into the same subset. A simple algorithm uses a set of vectors as the base. Then for each input vector, it finds the nearest base vector and uses inner product to move the base vector slightly toward that input vector. After a few iterations over the input set, the base vectors converge into a position that would best identify the division of the domain. Given another input vector, the classification would be fairly easy, as it only needs to find the nearest base vector.

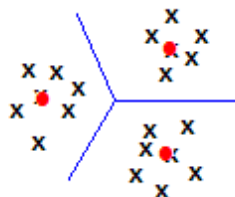


Figure 1 Vector Quantization in a two dimensional space. Crosses are input vectors and dots are base vectors

Behavioral partitioning uses the same idea:

1. Train each expert with a few random samples from the training data.
2. For each pair (\mathbf{x}, \mathbf{y}) in the training data:
 - 2.1. Feed \mathbf{x} to all experts and get the result \mathbf{y}_i of each expert.
 - 2.2. Find the nearest results to the target \mathbf{y} and train the corresponding expert with the pair (\mathbf{x}, \mathbf{y}) .
3. Repeat step 2 for n times.

Figure 2 Behavioral Partitioning

This method has been used and studied in a few previous works [7][9][10]. The expert type and training mechanism is not of much importance. The only constraint is that the training mechanism should be an on-line one, which means that the expert should be trained in an iterative manner. MLP neural networks are good choices for such domain [9][10].

This basic method can be further enhanced by introducing soft margins that is instead of training only the best expert, k best experts are trained [10].

The idea behind behavioral partitioning is that it introduces locality into expert domains. Instead of training all the experts, only the best working ones are trained. Therefore, every expert is only trained in the subset that it best works on. There might be variation of the method that try to train all the experts, but each one is only trained partially with a learning factor relative to it's fitness toward the sample input.

2.2. Decision Fusion

Once the experts are trained using the method described above, the fusion mechanism can be trained separately. Here, we shall make an assumption that the above method results in an optimum division. The fusion mechanism should be a selector, that given an input, would choose an expert that would produce the closest results to the actual answer. This is the place that the sample data can be used again. On the

other hand, training the selector is done using the following general algorithm:

1. For each pair (\mathbf{x}, \mathbf{y}) find the experts with the nearest result to the actual answer \mathbf{y} .
2. Train the selector to select that expert for the value \mathbf{x} .
3. Repeat steps 1 and 2 for n times.

Figure 3 Training the Fusion Mechanism

The main challenge with the above method is the fact that it might lead to over training. The method can be modified to add an early stopping mechanism, which would stop the selector from over training. Once again, the method can be enhanced by training the selector with the best k experts. The selector then would select one of the best experts given an input sample. This would also help with the problem of over training but will require that the classifier has a better generalization capacity. The classifier can be either a single model [7, 9] or a more complex ensemble model [10]. The model does not need to have the constraints as those the experts. Unlike the experts, it does not need to an on-line learning model. Any batch learning mechanism would suffice for the combiners. Support Vector Machine and other kernel methods are good choices for this case. Ensemble methods like HME structures are also good choices [10]. In an HME structure, all the internal nodes are classifier models. Unlike other HME structures, the learning model may be much more complex. As the experts are trained prior to the classification structures the optimization of the classification parameters are much simpler. In an HME structure, the general algorithm would be implemented as follows [10]:

1. For each pair (\mathbf{x}, \mathbf{y}) find the best experts.
2. Identify the path from the root of the hierarchy to the best expert and train only the gates on the path to select the right choice.
3. Repeat steps 1 and 2 for n times.

Figure 4 HME Selection Training Mechanism

Once again the method can be enhanced by training the gates on the path to the k best

experts. This would prevent the system from making wrong choices to some extent as it introduces soft margins for the selection. There are other ways to minimize the wrong selection risk. One way is to arrange the experts in the hierarchy in such a way that similar experts (in terms of localized behavior) would be close to each other [10]. In the case of a mistake in the classification, a similar expert to the optimum would be chosen which would result in a close answer to the desired one.

2.3. Ensemble in Action

Once the ensemble is trained with the above procedure, it can be easily used for regression purposes. Given an input \mathbf{x} , we first feed it into the selector (or the selection ensemble) which will choose an expert accordingly. Then we will feed the input to the selected expert and get the result.

It is important to notice that only one of the experts is active in this phase. Thus, even with complex learning models in the experts (e.g. MPL neural networks), the result can be calculated fast [9, 10]. Thus, the system can be used in a complex domain with high degree of nonlinearity.

3. Implementation Results

Implementations of this method have yielded satisfactory results. An ensemble of local experts with a single selector was used to compress images with MLP neural networks [9]. The work stated that the multi expert approach significantly outperformed single expert ones. A similar structure was used for the regression of multi-part functions [7].

A hierarchical version of the method was used for a set of regression problems defined on the DELVE [21] framework and was compared with similar learning methods. The results indicate that all the methods are useful in complex nonlinear domains where simple models did not produce good result.

The HME structure is tested again while the selectors are assumed to produce the optimum results. That is, given a testing set for each pair (x, y) the nearest result of the experts is selected and compared with the desired value. This way, the selection mechanism is bypassed to see how well the partitioning mechanism works. The test results on the DELVE framework indicate that even with small number of experts, this method outperforms most of the current ensemble learning methods. Although the selection mechanism fails to produce good results, the partitioning method seems to be a good one. Behavioral partitioning thus seems to be a good method for ensemble local learning.

4. Conclusion

The method presented in this article is a general mechanism for solving complex regression problems. The main idea works around breaking the training and selection mechanism into separate steps. To do so, we make use of a simple algorithm to behaviorally partition the problem space, after which we end up with a set of locally trained experts. Then we use the sample input data to train a selection mechanism to choose the correct expert for every input. The regression is then easily accomplished by selecting an expert for the input and using that expert to produce the result.

This method can also be applied to classification problems. The experts can be set to be classifiers, which are only trained, in a local subset of the problem domain.

Heterogeneous experts might also be useful as they can be trained in the subset they work best. This might be useful for problems with heterogeneous patterns in the domain where no single expert can be a good solution to the whole problem.

References

- [1] Waterhouse, S.R., "Classification and regression using mixtures of experts", Ph.D., Thesis, Department of Engineering, Cambridge University, 1997.
- [2] G. Valentini and F. Masulli, "Ensembles of learning machines", In M. Marinaro and R. Tagliaferri, editors, 13th Italian Workshop on Neural Nets, volume 2486 of Lecture Notes in Computer Science, pages 3--22. Springer-Verlag, 2002.
- [3] Bottou, L., & Vapnik, "Local learning algorithms", Neural computation, 4(6), 888-900.
- [4] Jordan, M.I., & Jacobs, R.A., "Hierarchical mixtures of experts and the EM algorithm", Neural Computation, 1994, 6, 181-214.
- [5] Hinton, G. E., B. Sallans and Z. Ghahramani, "A hierarchical community of experts", In: Learning in Graphical Models (M. I. Jordan, Ed.). 1998, pp. 479-494. Kluwer Academic Publishers.
- [6] M.I. Jordan and L. Xu, "Convergence results for the EM approach to mixtures of experts architectures", Neural Networks, 8:1409--1431, 1995.
- [7] Mahdi Keramati, "Competitive Behavioral Partitioning of the Input Space for Local Experts", ECE Symp, University of Tehran, 2003
- [8] R. Sun and T. Peterson, "Automatic partitioning for multi-agent reinforcement learning", From Animals to Animats: Proceedings of the International Conference of Simulation of Adaptive Behavior (SAB'2000). Paris, France. MIT Press, Cambridge, MA. 2000.
- [9] Mahdi Milani Fard, "A Coevolutionary Competitive Multi-expert System for Image Compression with Neural Networks", In Proc of IEEE Intl. Conf. on Engineering of Intelligent Systems, Pakistan, 2006.
- [10] Mahdi Milani Fard, A. Bakhtiary, "Behavioral Partitioning in a Hierarchical Mixture of Experts using K-Best-Experts Algorithm", submitted to IEEE Symposium on Foundations of Computational Intelligence (FOCI'07)
- [11] L.I. Kuncheva, "A Theoretical Study on Six Classifier Fusion Strategies", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pp. 281-286, Feb. 2002.
- [12] Huang, Y.S., Suen, C.Y., "A Method of Combining Multiple Classifiers: A Neural Network Approach", ICPR94(418-420). BibRef 9400
- [13] Giorgio Fumera, Fabio Roli, "A Theoretical and Experimental Analysis of Linear Combiners for Multiple Classifier Systems", IEEE Trans. Pattern Anal. Mach. Intell. 27(6): 942-956 (2005)
- [14] Giacinto, Roli F. Dynamic classifier selection based on multiple classifier behavior. Pattern Recognition, 2001,34 (9)
- [15] Marco F. Duarte and Yu-Hen Hu, "Decision Fusion in Collaborative Sensor Networks"
- [16] Kittler, J.V., Alkoot, F.M, "Sum versus vote fusion in multiple classifier systems", PAMI(25), No. 1, January 2003, pp. 110-115.
- [17] Luigi P. et-al, "Optimizing the Error/Reject Trade-Off for a Multi-Expert System Using the Bayesian Combining Rule", SSPR/SPR 1998: 716-725
- [18] Yea S. Huang, Ching Y. Suen, "A Method of Combining Multiple Experts for the Recognition of Unconstrained Handwritten Numerals", IEEE Trans. Pattern Anal. Mach. Intell. 17(1): 90-94 (1995)
- [19] V. Petridis, et-al, "A Bayesian Multiple Models Combination Method for Time Series Prediction", Journal of Intelligent and Robotic Systems, v.31 n.1-3, p.69-89, May -July 2001
- [20] Matthias Rychetsky, et-al, "Application of Hierarchical Mixture of Experts Networks to Engine Knock Detection", 5th European Congress on Intelligent Techniques and Soft Computing, September 08. - 11, 1997
- [21] DELVE - Data for Evaluating Learning in Valid Experiments. Developed at University of Toronto. <http://www.cs.toronto.edu/~delve/>