Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 96 (2016) 723 – 732

Procedia
Computer Science

20th International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2016, 5-7 September 2016, York, United Kingdom

Local learning for multi-layer, multi-component predictive system

Bassma Al-Jubouri[a], Bogdan Gabrys[a]

[a] *Bournemouth University, Fern Barrow, Poole, Dorset, Bournemouth, BH12 5BB, United Kingdom*

Abstract

This study introduces a new multi-layer multi-component ensemble. The components of this ensemble are trained locally on subsets of features for disjoint sets of data. The data instances are assigned to local regions using the similarity of their features pairwise squared correlation. Many ensemble methods encourage diversity among their base predictors by training them on different subsets of data or different subsets of features. In the proposed architecture the local regions contain disjoint sets of data and for this data only the most similar features are selected. The pairwise squared correlations of the features are used to weight the predictions of the ensemble's models. The proposed architecture has been tested on a number of data sets and its performance was compared to five benchmark algorithms. The results showed that the testing accuracy of the developed architecture is comparable to the rotation forest and is better than the other benchmark algorithms.

Peer-review under responsibility of KES International

Keywords: local learning; multi-layer multi-component predictive system; feature selection

1. Introduction

In the past decades ensemble learning has been an active area of research in machine learning. One of the earliest work on ensemble learning was introduces in[1], where the authors suggested partitioning the feature space using two or more classifiers. Later on Breiman introduced Bagging[2], a simple to implement method, where the base predictors of the ensembles are trained on bootstrap replicas of the training data. Schapire[3] proved that a strong learner (in probably approximately correct sense) can be built by combining a number of weak learners using Boosting algorithm. This had led to the development of AdaBoost, one of the most successful ensemble learning algorithms, and its many variations to solve multi-class and regression problems. In the past decade many well performing ensemble methods were developed and used in a wide area of applications, such as stacked generalization[4], mixture of experts[5] and consensus aggregation[6] among others.

In literature, it has been shown that using an ensemble of predictors can often improve the generalization performance compared to that of a single predictor. The conditions for this improvement is for the base predictors to be diverse with respect to their errors and that they have a reasonable performance level[7],[8],[9] and[10].

E-mail address: baljubouri@bournemouth.ac.uk

Peer-review under responsibility of KES International
doi:10.1016/j.procs.2016.08.256

In order to design diverse predictors a number of approaches can be used, such as: varying the initial conditions (e.g. start each predictor with different randomly generated weights)[11], varying model architecture or model type[12],[13],[14] and[15], or varying the training data (where the base predictors are trained on a subset of the training data[2] or a subset of the features[16]). The latter approach is more likely to generate diverse models than the previous two approaches[17]. Many well performing ensemble algorithms use sampling techniques either in the feature space (like in rotation forest[16]) or in the instance space (like in Bagging[2]) to generate their base predictors. This study aims to combine the two sampling techniques by training the base predictors on a subset of features for a subset of data. The locality of the selected features are determined based on the pairwise similarity of their squared correlation. A measure which was introduced in[18] and has been successfully used in image recognition task when applied with deep learning algorithms is adapted in the proposed algorithm.

Once the models have been trained, a fusion method is used to combine them into a single ensemble. The fusion method used in this study is Weighted Majority Vote (WMV)[19], where the weights are assigned to the values of the pairwise squared correlation of the features.

A two-layer, multi-component predictive system is presented in this paper. In the first layer the data is split according to their features pairwise similarity and is assigned to a set of Local Regions (LRs). For each LR a number of models are trained and an ensemble is constructed. In the second layer an ensemble that combines the ensembles outputs from the previous layer using WMV is constructed to provide the final prediction. The performance of this system is tested on several data sets and compared to five benchmark algorithms.

The organization of this paper is as follows: in the next section multi-layer, multi-component predictive systems are introduced, followed by a section that present sampling techniques and shed light on the pairwise similarity metric used in this work. In section four the experimental work will be discussed, this is followed by the obtained results in section five. Finally the conclusions, limitations and future work are given in section six.

2. Multi-layer, multi-component predictive systems

The use of multi-layer, multi-component predictive systems have shown many theoretical and practical benefits compared to the use of a single best model[20], these include: statistical benefits, as combining the output of several classifiers can often compensate for the possible unfortunate poor prediction of a single predictor, also, it is beneficial to use ensemble learning when the data is too large or too small. Furthermore, when the problem is too difficult to solve by a single predictor, ensemble methods can provide a divide and conquer strategy that a single predictor is incapable of achieving. Finally, when the data is generated from different resources (data fusion) a single predictor cannot represent the whole data accurately.

Once the base predictors of the ensemble are generated either a set or all of them are combined together. In general, combiners methods can be classified according to their ability to train (trainable vs. untrainable combiners) and to the type of their output (class label vs. continuous output combiners)[20 21]. One of the most widely used combining method is majority vote. It has been shown in[22] that by suitable organising of the predictive system into multi-component, multi-layer structure the limits of the majority vote error for such a system can be significantly expanded in comparison to a traditional single layer ensemble. The theoretical findings from[22] prompted the design of multistage selection fusion model discussed in[23] and subsequent very successful extensions and applications of multi-layer, multi-component systems in time series forecasting[24] and generic predictive modelling[25] with examples of applications to airlines ticket demand prediction[26], water pollution monitoring end prediction[27] or adaptable soft sensors development in process industry[25].

In this paper we continue our explorations of such systems. The multi-layer, multi-component predictive system used in this study is shown in Figure 1, where $w_{11},..., w_{1m}$, are the weights of the first layer and $w_{21},..., w_{2m}$ are the weights of the second layer. $M_1, ..., M_k$ are the base predictors of the first layer ensembles, $g_1, ..., g_m$ are the ensembles created from combining the base predictors and $\hat{Y}$ is the final prediction of the system. Let X be the data set containing the training objects, c represent the number of classes, Ω represent the actual class and $M_{i,k}$ represent the output prediction of the model, where $M_{i,k} = 1$ for class Ω_k and 0 otherwise and $k = 1, .., c$. The outputs of the base predictors $M_{i,c}$ and the ensemble g_j are given as c-dimensional binary vectors where $[M_{i,1}, .., M_{i,c}]^T \in \{0, 1\}^c$, $i = 1, .., k$ and $[g_{1,c}, .., g_{j,c}]^T \in [0, 1]^c$, $j = 1, .., m$ respectively. Equations 1 and 2 shows the mathematical representation for the

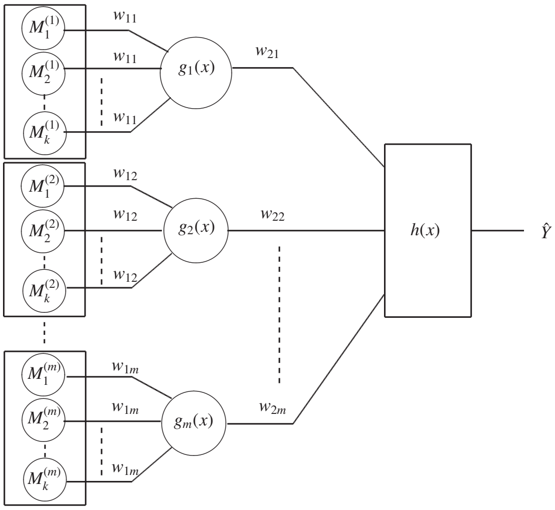

Fig. 1. The multi-layer multi-component predictive system.

ensembles generated from the first layer:

$$g_j(x) = \Sigma_{i=1}^{k} w_{1j} M_i^{(j)}(x) \tag{1}$$

and let

$$d_{j,k}(x) = \begin{cases} 1 & if\ g_j(x)\ labels\ x\ in\ \Omega_k, \\ 0 & otherwise. \end{cases} \tag{2}$$

Then in the second layer the ensemble is given by:

$$h(x) = \Sigma_{j=1}^{m} w_{2j} d_{j,k} \tag{3}$$

and the final prediction of the system is:

$$\hat{Y} = \arg\max_{c} h(x). \tag{4}$$

3. Generating diverse models

In order to generate diverse ensemble, the base predictors can be trained on either subsets of the data or subsets of the features. In the first approach the predictors do not see all of the training data which might be impractical in real scenarios were limited data is available. On the other hand, in the second approach a subset of the features are chosen for training each predictor. This can be achieved by using either feature extraction or feature selection methods[17]. In feature extraction methods, such as Principle Component Analysis (PCA) or Independent Component Analysis (ICA) the dimensionality of the data is reduced by creating new features that represent the projections of multiple existing features. In this case, the features selected for projection are the most discriminative features, thus there is a high probability that similar features will be chosen for creating the principle components. This can lead to training the base predictors on similar set of principle components, which eventually reduce the diversity among the predictors[28].

On the other hand, in feature selection methods, a different subset of features is chosen for the training of each base predictor. Subset selection can be achieved through many approaches. Some popular choices of feature selection in classification problems are correlation[10] and mutual information[29]. A recent study by[18] trains deep networks on local receptive fields that were created using a subset of features for a subset of data, where similar features are grouped together. The similarity of the features is measured using a variation of Pearsons product-moment coefficient. Pearson's correlation method can only show linear-dependencies between the features. In the work presented in[18] a measure for higher order dependencies between the features is used, this measure is the correlation between the energy responses of the features. The efficiency of this measure was illustrated in the results obtained when used with deep learning algorithm to solve unsupervised image recognition task.

Due to the success achieved using this similarity metric with deep learning network and due to the similarity between the architecture proposed in this study (shown in Figure 1) and deep learning network, this work was developed to use the squared correlation similarity metric to divide the data into LRs. The LRs data are used to train a set of local expert models. In order to generate the LRs, squared correlation between features is considered. Similar features are grouped in one region, such that the predictive model trained on the resultant subsets specialize in a particular aspect of the prediction problem. The chosen features are the ones with the highest correlation (but they are not identical). The high correlation between the features is viewed as an indication for their similarity in defining a certain region of the search space. On the other hand weakly correlated or independent features are assigned to different regions.

Mixing both sampling in the feature space as well as the instance space can encourage both diversity and locality of the resultant local models. Previous ensemble methods have shown improved performance when either one of these sampling methods were used. This study investigate the benefits of mixing those two approaches. The following subsection provides a detailed description of the correlation based local regions used in this work.

3.1. Correlation based local regions

This approach aims to group similar features into the same LR. The similarity metric used is the pairwise "*square correlation*" between the features (introduced in[18]). The reason for using squared correlation is that if the data set consists of linearly uncorrelated features, then a higher measure of correlation between the features can be found by computing the energy correlation between two features at a time (squared response).

Given a data set X that consists of linearly uncorrelated features (which can be obtained by applying a whitening procedure). Here we define x_j and x_k as two features in the data set. If $\mathbb{E}[x] = 0$ and $\mathbb{E}[(xx)^T] = I$, then the following similarity measure between the squared responses of features can be defined:

$$\begin{aligned} S[x_j, x_k] &= corr(x_j^2, x_k^2) \\ &= \mathbb{E}[x_j^2, x_k^2 - 1] / \sqrt{\mathbb{E}[x_j^4 - 1][x_k^4 - 1]} \\ &\equiv \frac{\sum_i x_j^{(i)^2} x_k^{(i)^2} - 1}{\sqrt{\sum_i (x_j^{(i)^4} - 1) \sum_i (x_k^{(i)^4} - 1)}} \end{aligned} \tag{5}$$

where i represent the sample index.
The following points summarise the steps used to generate the local regions using this metric:

- Whiten the input data set using Zero-phase Component Analysis (ZCA) whitening[30].
- Compute the pairwise similarity between all the features using Equation 5.
- Select N rows, $j_1, ..., j_N$ of the similarity matrix S.
- Construct LRs containing the top M values for the N rows of $S_{j,k}$

Each one of the N rows serves as a seed for a single region. Once the seeds for the LRs are chosen, the pairwise squared correlation for the features of each new instance is computed and compared with the seeds of the LRs. The instances are assigned to the local regions with the highest similarity. At the end of this stage an N disjoint sets (LRs) are constructed. Based on the LRs found, a multi-layer, multi-model predictive system is built.

4. Methodology

The procedure used to construct the multi-layer predictive system presented in this work encompasses the following phases: a) data preparation and partitioning, b) model generation and c) ensemble combination. In the following subsections a detailed description of these phases is given.

In order to validate and examine the generalization ability of the proposed architecture, Density Preserving Sampling (DPS)[31] is used to partition the data. DPS divide the data into subsets that are guaranteed to be representative of

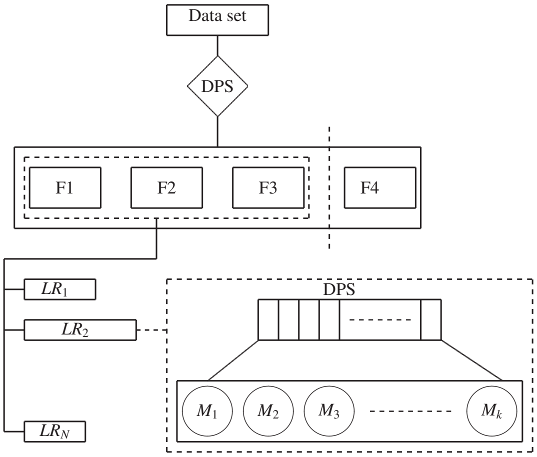

Fig. 2. Data preparation and model generation.

the whole data set[31]. In this work DPS is used to split the data into training and testing sets and then it is used to split the LRs data into K folds. K models are trained on the generated folds. Initially both DPS and Cross Validation (CV) were used to split the LRs data. However, when comparing the accuracies obtained from the two methods it was found that the models trained on CV folds had large variation in their accuracies. On the other hand, the models that were trained on the folds generated from the DPS were more stable (i.e. they had low variance in their error estimation). Due to this, DPS will be used in this work to partition the data.

4.1. Data preparation and partitioning

After loading the data the following procedure is used to preprocess and partition the data. The data goes through three partitioning stages, first the whole data is split into training and testing sets, then the training set is allocated to the LRs and finally within the LRs the data is split into K subsets which are used to train the local models. Figure 2 shows the preparation and partitioning of the data, where, $F_1, ...F_4$ are the folds generated from the first DPS split, $LR_1, ...LR_N$ are the LRs and $M_1, ...M_k$ are the local models within the regions trained using data from the second DPS split.

The points given below summarise the procedure used in this phase:

- Apply DPS to split the data into 4 representative folds.
- Use 3 out of 4 folds as the training data and the last fold as the testing data. Repeat for all four folds, so that each time different fold is used for the testing.
- Whiten the features of both the training and testing data using ZCA.
- Compute the pairwise squared correlation (energy) among all of the features of the training data and store the results in the similarity matrix.
- Choose N rows from the similarity matrix to be the seeds for the LRs.
- Compute the pairwise squared correlation for the features of each training instance and compare the results with the seeds of the N LRs. Add the instance to the LR that has the highest similarity with respect to its feature values.
- Apply *k* fold DPS to the LRs data.

Note that applying the whitening procedure to the data remove the linear dependencies within the data. In such case a measure of a high order dependencies between two features can be obtained by looking at the correlation of their energy (squared responses).

Table 1. Weights calculation for the illustrative example.

f1	f2	f3	f4	f5	f6	f7	LRs weights
C1		C3		C5	C6		$C1 + C3 + C5 + C6$
	C2	C3			C6	C7	$C2 + C3 + C6 + C7$
C1	C2		C4			C7	$C1 + C2 + C4 + C7$

4.2. *Model generation, testing and combining*

Once the data is assigned to the relevant LRs, the second DPS is applied to generate the K folds within the LRs and K models are trained on the LR folds. Furthermore, for all new instances N weights values are computed with respect to the N LRs. The following points give a detailed description for this phase:

- Train a predictive model on each of the K LRs folds.
- Compute the pairwise squared correlation for each testing data instance and measure its similarity to the seeds of the LRs. The summation of the energy for the corresponding features (to the LR) is used to weight the predictions of the LRs models. Given bellow is an illustrative example on how the weights of the LRs are calculated for a single instances.
  Where $f1...f7$ are the features of the samples and $C1...C7$ are the pairwise squared correlation of the features. When a new sample arrive the values $C1...C7$ are computed and the summation of the selected features (with respect to the LRs) is used as weights for the corresponding LR's prediction.

In the first layer, ensembles are generated from combining the models of the individual LRs, while, in the second layer an ensemble of the first layer ensembles is generated. The combining methods used is a WMV with the pairwise squared correlation of the features used as the weights in both layers.

5. Results

The Local learning Multi-layered (LLM) architecture described in the previous section is applied to the data sets shown in Table 2. The data sets used are taken from the UCI machine learning archive[32]. In this study the models used to provide the predictions will be referred to as base predictors. The base predictors used in the first set of experiments are CART Decision Trees (DTs) and in the second set of experiments are feed forward Neural Networks (NNs). The testing accuracies of the proposed architecture are compared to the accuracies of five benchmark algorithms, these are: Rotation Forest (RF), AdaBoost, Bagging, single CART DT and Linear Discriminant Analysis (LDA).

The setting of the algorithms used in these experiments are given bellow. A predefined number of LRs and number of models within each LR is selected. The parameters are chosen for illustration purpose and so that the results obtained can be compared across all the data sets. Also, it highlights the advantages and drawbacks of predefining these parameters with respect to the data set size and dimensionality:

- LLM: 6 LR's are used each have 8 models (48 DT's in total) trained on disjoint subsets of the data. The number of features used in the LRs is determined through a separate optimization routine. where four different numbers of features are considered (with step size equal to the number of features/4) and the number of features that generate the maximum testing accuracy is chosen.
- RF: the number of classifier are 6 and the number of disjoint features subspaces are 6.
- AdaBoost and Bagging: 60 DT were used as the weak learner for both algorithms, this value is chosen to be slightly higher than the number of DT's used in the LLM.
- Single DT and LDA: a single CART DT and a linear discriminant analysis.

Table 2. Data sets used in the experiments.

Data sets	Examples	Features	Classes
Spambase	4601	57	2
German credit cards	1000	20	2
Gaussian 8D	5000	8	2
Pima Indians Diabetes	768	8	2
Ionosphere	351	34	2
Vehicle	846	18	4
Waveform	5000	40	3

The following subsection discusses the internal accuracies of the proposed architecture and the comparison of the testing accuracies with the benchmark algorithms. This is followed by another subsection that investigates the effect of changing the base predictors model type on the performance of both the RF and the LLM architecture.

5.1. Internal Accuracies and Benchmark Comparison

The original data sets are split using DPS into four folds, and each time a different fold is used for testing and the remaining three folds are used for training. The performance of the LRs models can vary over the four iterations and in most cases there is no single LR that dominates the others over the four iterations. Changing one fold of the training data can affect the performance of the LRs, where it can become better or worse and/or have a smaller or larger variation in its models accuracies. This is due to the new data instances being assigned differently to the LRs based on their pairwise similarity with the regions seeds. The term internal accuracy in this work refer to the variation in the accuracies of the LRs base predictors. An example of the internal accuracies of the LRs for the Gaussian 8 dimensional data set is illustrate in Figure 3. It can be seen in Figure 3 that there is no single LR's that has the best accuracy over the four iterations. Also the amount of variation in the accuracies of the individual LRs changes over the four folds, for example in iteration 2 the 6*th* LR has the smallest variation among its 8 models accuracies, however, for the same LR the amount of variation is higher in the other iterations. The same behaviour can be seen in the internal accuracies of the other data sets.

In general, large data sets lead to more stable LRs and reduce the accuracy variation within and across the LRs. However, how big a data set should be, depends on the dimensionality of the data and its distribution in the search space. A small data set with high dimensionality like the ionosphere data set can have a wide range of variation in the LRs accuracies. This is due to the small number of instances that are assigned to the LRs, which can lead to a model being validated on few or a single data instance(s). Classifying this instance correctly or incorrectly results in 0% or 100% accuracy. The solution to this problem is to lower the number of LRs and/or the number of their internal models. However, in order to be able to compare the accuracies over all of the data sets used in these experiments, the same number of LRs and models were used.

Benchmark Comparison

The overall testing accuracies of the LLM architecture are evaluated and compared with the five benchmark algorithms described above. Table 3 shows the average testing accuracy for the learning algorithms over the four DPS folds. The LLM architecture has the highest testing accuracies on three data sets (Gaussian 8D, Pima Indians Diabetes and vehicle). It perform very closely to the RF on Spambase (with 0.30% difference) and has a slightly lower testing accuracy than Adaboost on the German credit card data set (less than 0.2%). Also, it is the second best after the RF on the Ionosphere data set (as was mentioned before this is due to the small number of instances and the high dimensionality of the Ionosphere data set).

Table 3. Benchmark Comparison: Testing accuracy.

Data sets	LLM	RF	Adaboost	Bagging	singleDT	LDA
Gaussian 8D	88.16	80.70	69.94	63.68	50	51.68
German credit cards	70	65.30	70.20	49.80	56	54.10
Ionosphere	77.19	92.61	74.64	62.09	56.99	67.53
Spambase	85.20	85.50	63.70	67.05	60.60	42.21
Pima Indians Diabetes	76.62	73.3	35.03	36.59	65.10	34.90
Vehicle	67.61	61.37	25.65	25.53	25.41	28.25
Waveform	65.68	91.46	42.46	61.24	49.84	65.92

5.2. *Changing the type of the base predictors*

In this subsection the type of the base predictors was changed from CART DTs to feed forward NNs for both the LLM architecture and the RF method. The accuracies of both methods using the new base predictor is shown in Table 4. In general, changing the base predictors to NNs improves the overall testing accuracies for both methods. However, the improvement in the testing accuracy of the RF method, when NNs is used as the base predictors, is higher than that of the LLM architecture. Furthermore, the test accuracy of the LLM architecture has improved on all but the ionosphere data set and the Pima Indians Diabetes, where the it has deteriorated by 2.94% ans 0.3% respectively. It remains the same on the German credit card.

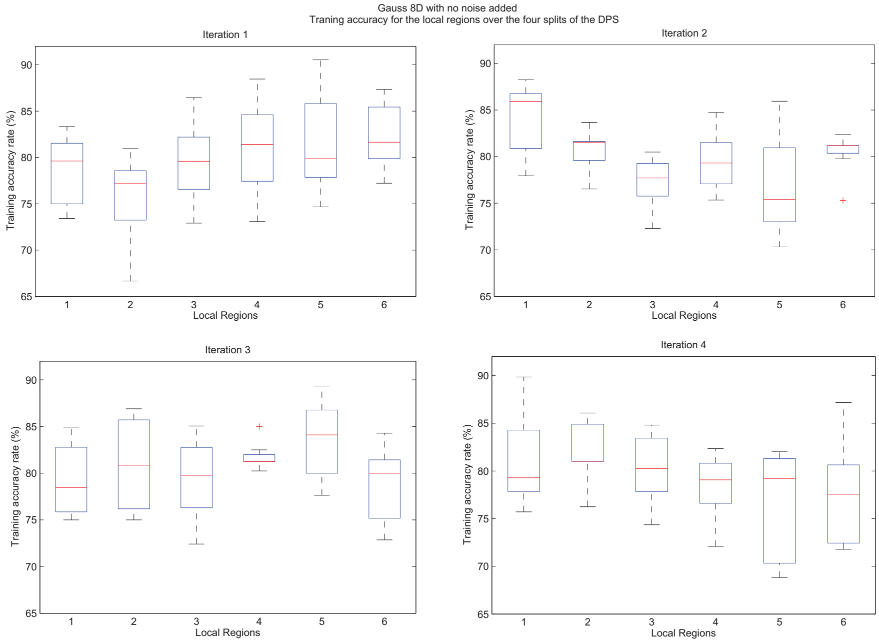
Gauss 8D with no noise added
Traning accuracy for the local regions over the four splits of the DPS
Iteration 1
Iteration 2
Iteration 3
Iteration 4

Fig. 3. Training accuracies of the local regions models for the Gaussian 8D data set.

Table 4. Benchmark Comparison: Testing accuracy using NNs as the base predictors

Data sets	LLM test	RF test
Gaussian 8D	88.45	88.4
German credit cards	70	70
Ionosphere	74.25	93.15
Spambase	90.55	85.75
Pima Indians Diabetes	76.30	76.8
Vehicle	78.50	81.75
Waveform	85.05	92.65

5.3. Further discussion

The LLM architecture has been tested on a number of data sets. The results showed that there are different levels of variation in the accuracy among the LRs models. The amount of this variation is affected mainly by the size and dimensionality of the data. If there is enough data to train and validate the LRs models, the variation is small and if the data is not enough the variation is high. The architecture with the current setting works best for large data sets, as can be seen from the testing accuracies of the architecture when applied to data sets like Gaussian 8D or Spambase. However, the results showed that both the number of LRs and number of models developed within the LRs need to be optimised in accordance with the data set size and dimensionality. Changing the base predictors to NNs had a better effect on the prediction accuracy of the RF method than it had on the LLM architecture. There is an increase in the prediction accuracy of both algorithms; however, the increase in the RF accuracy is higher than that in the LLM.

6. Conclusions, limitations and future work

This study introduced a local learning based algorithm for multi-layer, multi-component architecture. The architecture consists of multiple LRs, each of which has multiple models developed on subsets or all of the features for disjoint sets of data. The way in which the features are selected and assigned to the individual LRs depends on the similarities of their pairwise squared correlation. This generation method of the LR can be applied in supervised as well as unsupervised learning as it does not consider the output class when splitting the data.

The results show that the overall testing accuracies of the proposed architecture exceeded the average internal accuracies of the LRs models. This is due to the fact that the LRs are trained on disjoint sets of the data. However, as the prediction of the LRs is weighted by the squared correlation of their features, a higher degree of importance is given to the prediction of the most similar LR(s) and this can often lead to the correct prediction.

In order to create an ensemble, two aspects should be taken into consideration: how the base predictors are generated and how they are combined. This study investigated one aspect of the problem that is the generation of diverse models; however, it does not focus on how these models are combined. Future work will investigate the use of different types of combiners and examine the effect it can have on the performance of the proposed architecture. Furthermore, one of the advantages of using ensembles with diverse models is their robustness to noise in the data. Further work will test the proposed architecture on data with added level of noise and examine the complementary behaviour of its models.

Acknowledgements

The research leading to these results has received funding from the European Commission within the Marie Curie Industry and Academia Partnerships & Pathways (IAPP) programme under grant agreement *no.*251617.

References

1. Dasarathy, B.V., Sheela, B.V.. A composite classifier system design: concepts and methodology. *Proceedings of the IEEE* 1979;**67**(5):708–713.
2. Breiman, L.. Bagging predictors. *Machine learning* 1996;**24**(2):123–140.
3. Schapire, R.E.. The strength of weak learnability. *Machine learning* 1990;**5**(2):197–227.
4. Wolpert, D.H.. Stacked generalization. *Neural networks* 1992;**5**(2):241–259.
5. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.. Adaptive mixtures of local experts. *Neural computation* 1991;**3**(1):79–87.
6. Benediktsson, J.A., Swain, P.H.. Consensus theoretic classification methods. *Systems, Man and Cybernetics, IEEE Transactions on* 1992; **22**(4):688–704.
7. Jacobs, R.A.. Methods for combining experts' probability assessments. *Neural computation* 1995;**7**(5):867–888.
8. Meir, R.. Bias, variance and the combination of estimators; the case of linear least squares. In: *In Advances in Neural Information Processing Systems 7*. Citeseer; 1995, .
9. Opitz, D.W., Shavlik, J.W.. Actively searching for an effective neural network ensemble. *Connection Science* 1996;**8**(3-4):337–354.
10. Tumer, K., Ghosh, J.. Error correlation and error reduction in ensemble classifiers. *Connection science* 1996;**8**(3-4):385–404.
11. Brown, G., Wyatt, J., Harris, R., Yao, X.. Diversity creation methods: a survey and categorisation. *Information Fusion* 2005;**6**(1):5–20.
12. Islam, M.M., Yao, X., Murase, K.. A constructive algorithm for training cooperative neural network ensembles. *Neural Networks, IEEE Transactions on* 2003;**14**(4):820–834.
13. Opitz, D.W., Shavlik, J.W., et al. Generating accurate and diverse members of a neural-network ensemble. *Advances in neural information processing systems* 1996;:535–541.
14. Wang, W., Jones, P., Partridge, D.. Diversity between neural networks and decision trees for building multiple classifier systems. In: *Multiple Classifier Systems*. Springer; 2000, p. 240–249.
15. Langdon, W.B., Barrett, S., Buxton, B.F.. Combining decision trees and neural networks for drug discovery. In: *Genetic Programming*. Springer; 2002, p. 60–70.
16. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.. Rotation forest: A new classifier ensemble method. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 2006;**28**(10):1619–1630.
17. Xue, F., Subbu, R., Bonissone, P.. Locally weighted fusion of multiple predictive models. In: *Neural Networks, 2006. IJCNN'06. International Joint Conference on*. IEEE; 2006, p. 2137–2143.
18. Coates, A., Ng, A.Y.. Selecting receptive fields in deep networks. In: *Advances in Neural Information Processing Systems*. 2011, p. 2528–2536.
19. James, G.. *Majority vote classifiers: theory and applications*. Ph.D. thesis; Stanford University; 1998.
20. Polikar, R.. Ensemble based systems in decision making. *Circuits and systems magazine, IEEE* 2006;**6**(3):21–45.
21. Ruta, D., Gabrys, B.. An overview of classifier fusion methods. *Computing and Information systems* 2000;**7**(1):1–10.
22. Ruta, D., Gabrys, B.. A theoretical analysis of the limits of majority voting errors for multiple classifier systems. *Pattern Analysis & Applications* 2002;**5**(4):333–350.
23. Ruta, D., Gabrys, B.. Classifier selection for majority voting. *Information fusion* 2005;**6**(1):63–81.
24. Ruta, D., Gabrys, B., Lemke, C.. A generic multilevel architecture for time series prediction. *Knowledge and Data Engineering, IEEE Transactions on* 2011;**23**(3):350–359.
25. Kadlec, P., Gabrys, B.. Architecture for development of adaptive on-line prediction models. *Memetic Computing* 2009;**1**(4):241–269.
26. Riedel, S., Gabrys, B.. Pooling for combination of multilevel forecasts. *Knowledge and Data Engineering, IEEE Transactions on* 2009; **21**(12):1753–176.
27. Budka, M., Gabrys, B., Ravagnan, E.. Robust predictive modelling of water pollution using biomarker data. *Water research* 2010; **44**(10):3294–3308.
28. Tumer, K., Oza, N.C.. Input decimated ensembles. *Pattern Analysis & Applications* 2003;**6**(1):65–77.
29. Cover, T.M., Thomas, J.A.. *Elements of information theory*. John Wiley & Sons; 2012.
30. Bell, A.J., Sejnowski, T.J.. The independent components of natural scenes are edge filters. *Vision research* 1997;**37**(23):3327–3338.
31. Budka, M., Gabrys, B.. Density-preserving sampling: robust and efficient alternative to cross-validation for error estimation. *Neural Networks and Learning Systems, IEEE Transactions on* 2013;**24**(1):22–34.
32. Lichman, M.. UCI machine learning repository. 2013. URL: `http://archive.ics.uci.edu/ml`.