Introducing the Theory of Probabilistic Hierarchical Learning for Classification

Ziauddin Ursani(✉) and Jo Dicks

Quadram Institute Bioscience, Norwich, UK
ziauddin.ursani@quadram.ac.uk

Abstract. This is the 5th paper in our series of papers on hierarchical learning for classification. Hierarchical learning for classification is an automated method of creating hierarchy list of learnt models that are on the one hand capable of partitioning the training set into equal number of subsets and on the other hand are also capable of classifying elements of each corresponding subset into classes of the problem. In this paper, the probabilistic hierarchical learning for classification has been formalized and presented as a theory. The theory asserts that the accurate models of complex datasets can be produced through hierarchical application of low complexity models. The theory is validated through experiments on five popular real-world datasets. Generalizing ability of the theory is also tested. Comparison with the contemporary literature points towards promising future for this theory. The theory is covered by four postulates, which are carved out elegantly through mathematical formalisms.

Keywords: Hierarchical learning · Probabilistic learning · Set-partitioning

1 Introduction

We have set this introduction to differentiate between hierarchical learning for classification and hierarchical classification itself. The word hierarchical classification has been used in numerous contexts therefore, we have designed this introduction in a way to exclude the irrelevant contexts hierarchically one by one to mark the constrained field of theory of probabilistic hierarchical leaning for classification.

The theory of probabilistic hierarchical leaning for classification is not about hierarchical classification analytically done by human beings. The most profound example of this is the classification of all biological organisms on earth e.g. [1]. The biological organisms are now classified into eight levels i.e., domains, kingdoms, phyla, classes, orders, families, genera and lastly into species in that hierarchical order. This hierarchical classification can be represented through a directed acyclic graph (DAG) e.g. [2]. While considering DAG representation of biological classes, domains can be placed at the root node while species can be placed at the leaf node.

The theory of probabilistic hierarchical leaning for classification is not about hierarchical classification using computational learning methods, where hierarchies are decided meticulously by humans themselves. The classical example of this is a

F. Wotawa et al. (Eds.): IEA/AIE 2019, LNAI 11606, pp. 628–641, 2019.
https://doi.org/10.1007/978-3-030-22999-3_54

Hierarchical Support Vector Machines (H-SVM) [3]. In these methods hierarchies are not set by computers but decided prior to start of a computer program. This is done by merging elements of several classes into one meta-class then applying SVM as a binary classifier between one class against a meta-class. Then again in the next hierarchy another class is extracted from the meta-class and SVM is applied to classify between the newly extracted class against a remaining meta-class. This procedure continues until meta-class retains elements from only one class of the dataset.

The theory of probabilistic hierarchical leaning for classification is not about automated generation of meta-classes either. The automated generation of meta-classes was proposed in 2008 for a handwriting character recognition system [4]. However, to our understanding creation of a meta-class is an artificial creation of class hierarchy where the actual classes are flat not hierarchical.

The theory of probabilistic hierarchical leaning for classification is not about hierarchical classes at all. This theory proposes a model of hierarchical learning even though classes of the dataset are flat. The model of hierarchical learning consists of hierarchy of learnt models rather than hierarchy of classes. The model in each hierarchy is applicable to a subset of the training set created during training in the corresponding hierarchy. Please note that subset created in a hierarchy doesn't represent a single class or a meta-class containing several classes. This is just a subset of the training set containing some of the elements from various classes. Therefore, this subset doesn't represent class hierarchy. This only represents hierarchy of learning where both the model and its area of influence are learnt altogether [5–8].

One might argue that theory of probabilistic hierarchical learning is similar to ensemble learning [9] because both contain multiple models. However, this is an inaccurate assessment. This is because unlike theory of probabilistic hierarchical learning, models in the ensemble learning are not hierarchically learnt. Furthermore, in ensemble learning domain of each of the models covers whole training set, whereas in the theory of probabilistic hierarchical learning sum of domains of all the models is equal to one training set. Therefore, method based on theory of probabilistic hierarchical learning can be much quicker than ensemble learning. Additionally, ensemble learning consists of averaging error of all the models over the whole training set in contravention of the theory of probabilistic hierarchical learning where learnt models are error free in their constrained domains. Finally, models in the ensemble learning are independent of each other therefore they can be applied simultaneously in parallel. However, this is not the case with the theory of probabilistic hierarchical learning where models are actually sub-models of a supermodel in a way that each sub-model is placed at one hierarchy of the supermodel. Therefore, these models can only be applied sequentially or hierarchically on their turn to the rest of the unclassified training set but not applied in parallel to the whole training set.

This paper is structured as follows. In Sect. 2 the theory addresses the question of why hierarchical learning in first place? The theory proposes the model of probabilistic hierarchical learning in Sect. 3. In Sect. 4 the theory sets out the probability of class membership as a corner stone of hierarchical learning. The alternative methodology for class membership under specific circumstances is discussed in Sect. 5. The litmus test of the theory is designed and experimented in Sect. 6. In Sect. 7, generalization ability of theory is experimented. Comparison of results with the literature is done in Sect. 8. Finally, in Sect. 9, conclusions are made, and future work is set out.

2 Hierarchical Learning-Why?

This section sets out very premise of the theory i.e., the "Hierarchical Learning". Why the hierarchical learning in first place? Response to this question is not very difficult to formulate. Complex problems require complex solution methodologies. Since we are classifying the complex datasets, so they require complex solution methodologies. Such complex methodologies are already present in the literature such as Deep Learning e.g. [10, 11] and Recurrent Neural Network e.g. [12]. In these methods we have several hidden layers for training the network. This is because one hidden layer is not enough to grasp the complexity of the problem. In pursuit of our search we have two objectives at hand. One objective is reducing complexity of our model and another objective is increasing its accuracy. Both the objectives are conflicting to each other. As we try to improve on accuracy, we make our model more complex. Therefore, any low complexity discriminant model is impossible to classify any meaningful real-world datasets of practical size. If we try to improve on accuracy with a discriminant model then we will be introducing more and more mathematical operators, which may end up in a very complex discriminant containing several mathematical operators and set of those operators would be very difficult or even impossible to generalize over wide spectrum of datasets. So, what if we keep our discriminant simply restricted to only four elementary mathematical operators $+, -, \times \& \div$? We should not expect from such a low complexity discriminant to classify the whole dataset. However, we can expect from such a discriminant that it may classify only a subset of the training set. If our expectation is reasonable, then this generates an idea. The idea is why not create multiple low complexity discriminants each for specific subset of the training set? In this paper, we have explored this idea. After a rigorous experimentation, we came to conclusion that the only way to materialize such an idea is the development of hierarchical learning procedure where in each hierarchy a model and its corresponding area of influence (subset) be learnt simultaneously. This scenario is depicted in Fig. 1.

It can be seen in Fig. 1 that the model M_1 divides the training set into subsets S_1 and S_1^u, then S_1^u is further divided by model M_2 into subsets S_2 and S_2^u and so on and finally the model M_{n-1} divides the remaining training set into subsets S_{n-1} and S_{n-1}^u. The subset S_{n-1}^{u} turns out to be equal to subset S_n, as no further division of this subset is needed. This is because it contains the elements belonging to only one class therefore there is no need of another trained model M_n. The scenario in the Fig. 1 can be generalised as a hierarchical model as shown in the Eq. 1.

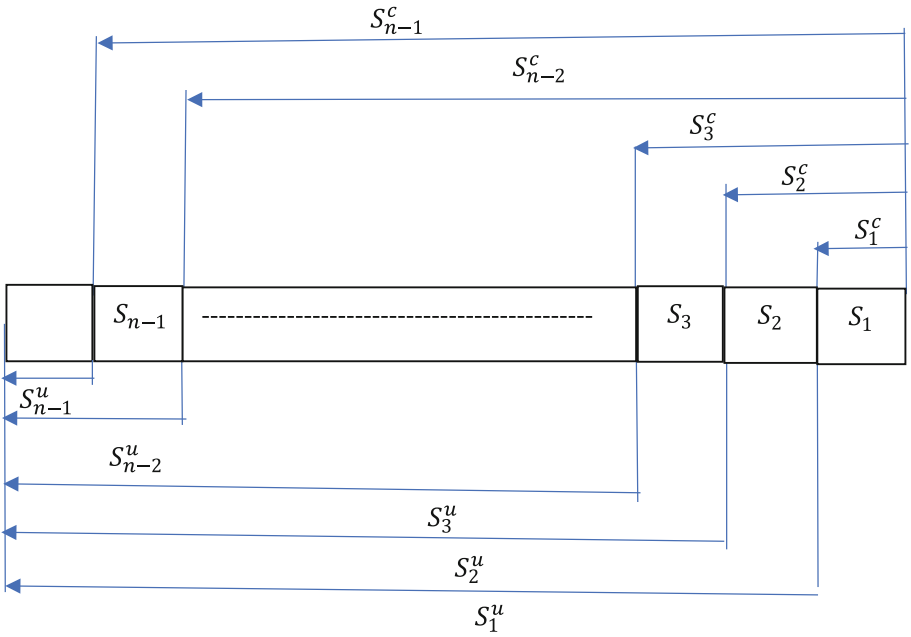

Fig. 1. Successive bifurcation of training set through hierarchical training of low complexity nonlinear discriminants.

$$\forall_{i\in U} H_i : \begin{cases} M_{i-1} \prec M_i : S_i \rightarrow C \\ U = S_i^c \cup S_i^u \\ S_i^c = \cup_{1\le j\le i} S_j \\ S_i^u = \left\{ \cup_{1\le j\le i} S_j \right\}' \end{cases} \tag{1}$$

Where

H_i = Hierarchy level i
M_i = Trained Model at hierarchy level i
S_i = Subset of training set at hierarchy level i
C = Class set
U = Training Set
S_i^c = Set of classified samples at hierarchy level i
S_i^u = Set of unclassified samples at hierarchy level i

The Eq. 1, says that

- at any hierarchy level i, model M_{i-1} precedes model M_i, whose domain is subset S_i and codomain is class set C
- at any hierarchy level i, the training set U is the union of classified S_i^c and unclassified S_i^u samples

- at any hierarchy level i, the classified set S_i^c is the union of all classified subsets preceding and including subset i
- at any hierarchy level i, the unclassified set S_i^u is the complement of the classified set S_i^c

It is emphasized that set of classified samples at any hierarchy level can contain data points from any number of available classes. From the above discussion following postulate can be formulated.

Postulate 1
High complexity model can be replaced with several low complexity models with constrained domains of the training set that could be trained one by one hierarchically until union of all constrained domains covers the whole training set.

3 Hierarchical Learning-How?

Now the question arises how the philosophy of postulate 1 could be materialized? If we have a close look at the model of hierarchical learning in the Fig. 1, we may start doubting the applicability of whole theory in first place. This is because it can be seen in the model in Eq. 1, the trained model in each hierarchy needs to achieve two-pronged classification in parallel i.e., categorization of elements within the corresponding subset into their original classes and also partitioning the remaining training set of unclassified elements into two subsets i.e. subset within its domain and subset outside its domain. There is no doubt as far as ability of model M_i to classify the elements within the subset S_i is concerned. This can normally be achieved using probability of class membership as shown in relation 2 below.

$$P(j) > \forall_{k \neq j} P(k) \Rightarrow j \in \{C_k\} \tag{2}$$

The relation 2 says that if the probability of the class membership of element j for class k is greater than its probability of class membership for each of the classes other than class k then the element j is the member of class k. This is the fundamental principle which most of the linear e.g. [13] or nonlinear e.g. [14] discriminants use for the classification. Now the question arises how the second part of classification could be achieved in parallel. If we look at objective of second part of classification carefully then we can make sense of it. Since the second part of classification involves partitioning of elements of training set into two subsets, one within and another outside the domain of the model M_i, therefore we need to decide which elements are within its domain. Naturally those elements which obey the relation (2) are within the domain of the model M_i. Therefore, we slightly modify the probabilistic model of relation 2 for hierarchical learning as shown in relation (3).

$$P(i,j,k) > \forall_{h \neq k} P(i,j,h) \Rightarrow j \in \{C_k, S_i\} \tag{3}$$

The relation 3 says that for model M_i if the probability of the class membership of element j for class k, i.e. $P(i,j,k)$ is greater than its probability of class membership for

each of the classes other than class k then the element j is the member of class k (C_k) and it is also the member of subset i (S_i), which is domain of model M_i. The elements which do not obey the probabilistic model of relation (3) are outside the domain of model M_i, as shown in relation (4).

$$P(i,j,k) > \forall_{h \neq k}\ P(i,j,h) \Rightarrow j \in \{C_h, S_i\} \tag{4}$$

The relation 4 says that for model M_i even though the probability of the class membership of element j for class k is greater than its probability of class membership for each of the classes other than class k but the element j is not the member of class k, it is the member of class h instead which is any class other than class k but it is still the member of subset S_i as a misclassified element, which is not desirable. Now the question arises through which mechanism the hierarchical learning could push element j out of subset S_i or domain of model M_i to avoid its misclassification. To understand this, we need to define a technical term 'Highest Misclassifying Margin' (HMM). The HMM is the greatest margin by which the model M_i, could misclassify a sample. The HMM can be calculated through Eq. 5.

$$\Delta_{max} = max\Big(\forall j \in \big\{C_h, S_i^u\big\} P(i,j,k) - max\big(\forall_{h \neq k} P(i,j,h)\big)\Big) \tag{5}$$

From Eq. 5, it can be seen that Δ_{max} (HMM) represents the maximum difference between the probabilities of wrongly assigned class and the maximum of probabilities from rest of the classes. Technically, we can incorporate HMM as computed in Eq. 5, in Eq. 4, to separate element j from subset S_i and thus prevent model M_i from misclassifying it, as shown in relation 6.

$$P(i,j,k) \not> \forall_{h \neq k}\ P(i,j,h) + \Delta_{max} \Rightarrow j \in S_i^u \tag{6}$$

It can be seen from relation 6, that probability of element j for the class k could not surpass the value on the right-hand side of the relation where Δ_{max} has been added to probabilities of the rest of the classes. This means element j is pushed out to subset S_i^u, which is not within the domain of model M_i and thus remains unclassified and should wait for next round of model training for the classification. However, it should be noted that element j could also be the member of right class C_k if the Δ_{max} was not introduced in the equation. This means that the Δ_{max}, not only pushes all the potentially misclassifying elements out of domain of model M_i but it does also push some of the potentially correctly classifying elements out of the domain of model M_i. However, with the introduction of Δ_{max} in the model, it is now confirmed that there will be no misclassification, either the element j will remain unclassified as in relation 6 or it will be classified correctly as in relation 7 below.

$$P(i,j,k) > \forall_{h \neq k}\ P(i,j,h) + \Delta_{max} \Rightarrow j \in \{C_k, S_i\} \tag{7}$$

By generalizing the relations (6–7) into one model we get

$$\begin{cases} j \in \{C_k, S_i\} & P(i,j,k) > \forall_{h \neq k} P(i,j,h) + \Delta_{max} \\ j \in S_i^u & otherwise \end{cases} \tag{8}$$

It should be noted that hierarchy i is the last hierarchy iff either $S_i^u = \{\emptyset\}$ or contains members belonging to one class only. We call it a remainder class. In any case whole training set is classified accurately. It should also be noted that if $S_i^u = \{\emptyset\}$ then $\Delta_{max} = 0$ else $\Delta_{max} > 0$. This means that there must always be some misclassifying margin if unclassified set is non-empty. It is emphasized that number of hierarchy levels is not fixed and entirely depends on the structure of the dataset and domain size of the models evolved.

Postulate 2
Misclassification of elements can be eliminated completely during hierarchical training with the incorporation of Highest Misclassifying Margin (HMM) in the fundamental model of probabilistic class membership, thus rendering the hierarchical training model error free.

4 Relative Closeness as Measure of Probability of Class Membership

From the probabilistic model (expression 8) presented in Sect. 3, it can be seen, that hierarchical learning is largely based on probability of class membership therefore it is better to call it probabilistic hierarchical learning. The probabilistic hierarchical learning could only be useful when computation of probability of class membership is easy, helpful and relevant. Now to understand this we need to think about how can we compute probability of element j for the membership of class C_k, with respect to model M_i, i.e., $P(i,j,k)$? It should be computed in a way that supports the probabilistic model (expression 8) in the objective of classification. In doing so, constraints associated with the notion of probability as a quantity could also be avoided and we will learn in a moment what we mean by that. Since the model is very simple therefore, the most natural way of computation of probability of class membership should be based on the relative closeness of element/sample to the mean of the class, i.e., closer the sample to the mean of the class greater should be the probability of its membership of the class. This can be understood from Eq. 9.

$$P_{(i,j,k)} = \begin{cases} \frac{\mu_{(i,j)} - \gamma_{(i,k,min)}}{\gamma_{(i,k,mean)} - \gamma_{(i,k,min)}}, & \mu_{(i,j)} \leq \gamma_{(i,k,mean)} \\ \frac{\gamma_{(i,k,max)} - \mu_{(i,j)}}{\gamma_{(i,k,max)} - \gamma_{(i,k,mean)}}, & otherwise \end{cases} \tag{9}$$

where

$\mu_{(i,j)}$ = value of sample j according to model M_i
$\gamma_{(i,k,min)}$ = estimated minimum value of samples of class C_k according to model M_i
$\gamma_{(i,k,max)}$ = estimated maximum value of samples of class C_k according to model M_i

$\gamma_{(j,k,mean)}$ = estimated mean value of samples of class C_k according to model M_i

The mean value of model M_i of class C_k can be estimated as follows.

$$\gamma_{(i,k,mean)} = \frac{\sum_{j \in C_k} \mu_{(i,j)}}{n_k} \tag{10}$$

where

n_k = Number of samples in the training set of class C_k

The maximum and minimum value among samples of class C_k according to model M_i can be estimated as,

$$\gamma_{\left(i,k,\begin{matrix} max \\ min \end{matrix}\right)} = \gamma_{(i,k,mean)} \pm 3.0^{*} \delta_{(i,k,sd)} \tag{11}$$

where

$\delta_{(i,k,sd)}$ = estimated standard deviation of samples of class C_k according to model M_i

The standard deviation of samples of class C_k according to model M_i can be estimated as,

$$\delta_{(i,k,sd)} = \frac{\sum_{j \in C_k} \left(\mu_{(i,j)} - \gamma_{(i,k,mean)}\right)^2}{n_k} \tag{12}$$

Now the model from Eqs. 9–12 suggests that the class C_k might have the class mean farther than the other classes from the sample j to whom it is assigned. This can be understood from the Fig. 2.

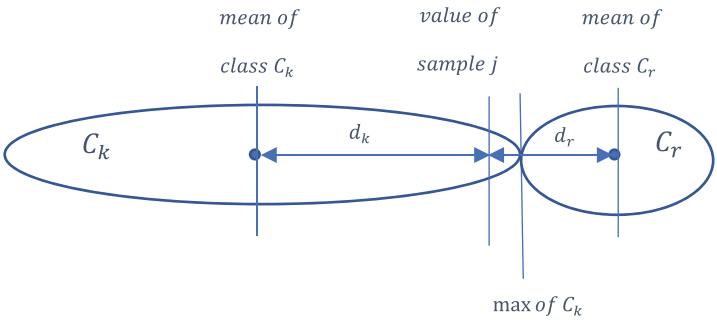

Fig. 2. Principle of relative closeness

It can be seen from Fig. 2, that $d_k > d_r$, but according to model in Eqs. 9–12 the sample will be assigned to class C_k instead of class C_r because value of sample j lies outside the boundary of class C_r but within the boundaries of class C_k. Therefore, even though value of sample j according to model M_i is closer to mean of the class C_r but it is assigned to class C_k because it is closer to class C_k in relative terms or in other words relatively closer to class C_k. However, the measure of relative closeness will only work well when we have good number of training samples belonging to each class, this is because it can be seen from Eq. 11 of the model that estimation of minimum and maximum of the class entirely depend on mean and standard deviation of the class sample values. These quantities are only meaningful when good number of samples are present in the training set.

Postulate 3
Relative closeness of sample to the mean of the class can be a useful measure for computation of probability of class membership when we have good representation of number of samples for each class in the training set.

5 Distance Inverse as a Measure of Probability of Class Membership

Now, since this is a hierarchical learning model, which bifurcates training set in each hierarchy into classified and unclassified samples therefore size of the training set continues to decrease with each hierarchy. In such a situation in the last hierarchy the training set may end up with very few samples such that number of samples of some or all the classes become less than 3. In such a case computation of standard deviation becomes meaningless and so estimation of minimum and maximum. To deal with this scenario, the measure of relative closeness is replaced with the measure of distance inverse. Therefore, Eq. 9 can be modified as Eq. 13 below.

$$P_{(i,j,k)} = \begin{cases} \frac{\mu_{(i,j)} - \gamma_{(i,k,min)}}{\gamma_{(i,k,mean)} - \gamma_{(i,k,min)}}, & \mu_{(i,j)} \leq \gamma_{(i,k,mean)}, \quad n_k \geq 3 \\ \frac{\gamma_{(i,k,max)} - \mu_{(i,j)}}{\gamma_{(i,k,max)} - \gamma_{(i,k,mean)}}, & otherwise, \quad n_k \geq 3 \\ \frac{1}{d_k}, & not\ applicable, \quad otherwise \end{cases} \tag{13}$$

where

n_k = number of samples in the training set for class C_k

It can be seen from the Eq. 13 that distance inverse measure is introduced when number of training samples of the class are less than 3. This changes measure of relative closeness to measure of closeness only or in popular terms measure of nearest neighbor.

Postulate 4
Closeness of sample to the mean of the class can be a useful measure for computation of probability of class membership when we have inadequate representation of number of samples for each class in the training set.

6 Litmus Test of the Theory

What is a litmus test that could validate the theory presented above? Expression 8 presents core model of the theory which is linked to postulate 2, which states that the misclassification can be eliminated completely. This means that learnt models should be able to accurately classify the training set. Therefore, if we can classify some of the popular datasets accurately through hierarchical learning of low complexity models, then this would mean that the basic idea behind the theory is valid. To see that the theory passes this litmus test we chose some of the popular real-world datasets from the UCI repository. The details of those datasets are tabulated in Tables 1 and 2. Table 1 gives feature description and Table 2 gives class description of each dataset.

Table 1. Feature description for each dataset

S. Nr.	Dataset	Nr. of features	Feature names
(1)	(2)	(3)	(4)
1	Iris flower	4	f_1: sepal length, f_2: sepal width, f_3: petal length, f_4: petal width
2	Balance scale	4	f_1: left weight, f_2: left distance, f_3: right weight, f_4: right distance
3	Car evaluation	6	f_1: buying cost, f_2: maintenance cost, f_3: number of doors, f_4: number of seats, f_5: size of lug-boot, f_6: level of safety
4	Banknote authentication	4	f_1: variance of wavelet transformed image (WTI), f_2: skewness of WTI, f_3: curtosis of WTI, f_4: entropy of image
5	Seeds	7	f_1: area, f_2: perimeter, f_3: compactness, f_4: length of kernel (k), f_5: width of k, f_6: asymmetry coeff:, f_7: length of k groove

We devised the training method [5–8] based on hierarchical learning theory above and coded in Microsoft Visual Studio C/C++. Please see details of the parameters and models learnt during training in our earlier works [5–8]. The program was applied on the five datasets described in Tables 1 and 2. The training method was applied for 30 simulations on each dataset on different random seeds. The trained models were then tested back on the same dataset. All the datasets were classified accurately in each simulation.

Table 2. Class description for each dataset

S. Nr.	Dataset	Nr. of classes	c_1	c_2	c_3	c_4	Total Nr. of samples
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
1	Iris flower	3	Setosa 50	Verginica 50	Versicolour 50	–	150
2	Balance scale	3	Balanced 49	Left tipped 288	Right tipped 288	–	625
3	Car evaluation	4	Unacceptable 1210	Acceptable 384	Good 69	Very good 65	1728
4	Banknote authentication	2	True 610	False 762	–	–	1372
5	Seeds	3	Kama 70	Rosa 70	Canadian 70	–	210

7 Generalizing Ability of the Theory

Retrieving accurate models of the complex datasets is an achievement but generalizing ability of such models should also be investigated. Generalizing ability means how such models perform on the unseen data or the data on which the model is not trained. We can devise experiments to test this ability of hierarchical learning. Let us develop hierarchical models on training sets containing only 50% randomly chosen samples of the original dataset and then test the model on the rest of the 50% samples on which they are not trained. The test results on this unseen data will show generalizing ability of the hierarchical model. To cross validate the models we reverse the roles of the training set and test set. Such an approach will reduce any statistical bias towards or against the hierarchical models. Furthermore, repeating this procedure for 30 independent runs will show close to average performance of the hierarchical learning theory. So, these experiments were performed on the same five datasets described in Sect. 6 and the results are reported in Table 3. In Table 3, column 1 shows serial number of the dataset, name of the dataset is given in column 2. Average results of 30 simulations are stated in column 3. Column 4 mentions best result in 30 simulations, column 5 provides percentage of accurate results in 30 simulations. This means the percentage of number of simulations out of 30 where 100% samples are correctly classified. Finally, column 6 just informs that whether data normalization procedure has been applied on the dataset. It can be seen from the results that in all the datasets more than 90% correct results have been obtained on average.

Table 3. Classification results

S. Nr.	Dataset	Average results	Best results	%age of accurate results	Data normalization
(1)	(2)	(3)	(4)	(5)	(6)
1	Iris	92.87%	94%	0.00%	No
2	Balance scale	99.11%	100%	3.33%	No
3	Car evaluation	93.09%	95.08%	0.00%	No
4	Banknote authentication	99.63%	99.93%	0.00%	No
5	Seeds	90.40%	93.33	0.00%	Yes

8 Comparison with State of Art

Now let us see how the results presented in Table 3 compare with the literature. For fair comparisons we need to compare this scheme with recently published methods that are applied on all the above datasets. We have chosen three recently published methods namely Support Vector Machines [15], Decision Trees [16] and random forest [17] that are applied on all the above five datasets. In Table 4 we compare the results of proposed approach with those methods. In Table 4, column 1 gives bibliographical reference, columns 2–6 state average results of five datasets. Number of simulations are mentioned in column 7, column 8 informs about x-validation type and finally size of the training set is revealed in column 9.

Table 4. Comparison with literature

Ref.	Iris	Bal. Scale	Car Evln.	Bn. Auth.	Seeds	Nr. of Sim.	x-valid	Size of Tr. Set
ID	%age	%age	%age	%age	%age	Int	type	%age
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
[15]	98.00	92.00	78.26	99.12	94.29	5	–	80.00
[16]	92.40	67.10	73.70	90.10	88.70	5	–	75.00
[17]	94.53	80.30	94.70	99.34	93.57	10	10-fld	90.00
Hierarchical learning	92.87	99.11	93.09	99.63	90.40	30	2-fld	50.00

It can be seen from the results that the proposed technique has outsmarted the three methods on the balance scale dataset with a very wide margin i.e. (99.11%/92.00%/80.30%/67.10%). The Hierarchical learning has also beaten the other three methods on the banknote authentication dataset, with smaller margins <1.00%. On the car evaluation dataset, the hierarchical learning has produced much better results than two methods (93.09%/78.26%/73.70) but little worse than the third method. On the rest of

the two datasets hierarchical learning has performed better than one of the techniques but worse than other two. However, it should be noted that hierarchical learning has used only 50% of the training set while the other three techniques have used 90%, 80% and 75% of training sets. Keeping this in mind the results produced by hierarchical learning can be regarded as respectable.

9 Conclusion and Future Work

This paper is fifth in our series of papers on hierarchical learning. This paper proposes the theory of probabilistic hierarchical learning covering four postulates. The first postulate says that multiple low complexity models can emulate the effect of high complexity model, when put together hierarchically. The second postulate says that chance of misclassification of the sample can be eliminated with smart use of fundamental model of probabilistic class membership. The third postulate proposes relative closeness rather than absolute nearness of sample to the mean of the class as basis for the probability of class membership. The fourth postulate proposes absolute nearness of sample to the mean of the class as basis for the probability of class membership in case of inadequate class representation in the training set. The theory is not only supported through mathematical analysis but also through experimentation on five popular classification datasets taken from UCI repository. In doing so, generalization ability of theory is also tested and compared with state of art showing satisfactory results. For this theory to work with large spectrum of datasets further theoretical enhancements are still needed which are currently under investigation.

Acknowledgement. Parts of this work were supported by the Biotechnology and Biological Sciences Research Council, through a Responsive Mode award (grant number BB/P022030/1) to J.D.

References

1. Ruggiero, M.A., et al.: A higher-level classification of all living organisms. PLoS ONE **10** (4), e0119248 (2015). https://doi.org/10.1371/journal.pone.0119248
2. Silla Jr., C.N., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Min. Knowl. Disc. **22**(1–2), 31–72 (2011)
3. Chen, Y., Crawford, M.M., Ghosh, J.: Integrating support vector machines in a hierarchical output space decomposition framework. In: Proceedings of the IEEE International Symposium on Geoscience and Remote Sensing, vol. 2, pp. 949–952 (2004)
4. Freitas, C.O.A., Oliveira, L.S., Aires, S.B.K., Bortolozzi, F.: Metaclasses and zoning mechanism applied to handwriting recognition. J. Univ. Comput. Sci. **14**(2), 211–223 (2008)
5. Ursani, Z., Corne, D.W.: Use of reliability engineering concepts in machine learning for classification. In: 4th International Conference on Soft Computing & Machine Intelligence (IEEE), (ISCMI 2017), Mauritius (2017)
6. Ursani, Z., Corne, D.W.: A novel nonlinear discriminant classifier trained by an evolutionary algorithm. In: 10th International Conference on Machine Learning and Computing (ICMLC 2018), 26–28 February 2018, University of Macau, China, ACM Conference Proceedings (2018). ISBN 978-1-4503-6353-2

7. Ursani, Z., Corne, D.W.: A hierarchical nonlinear discriminant classifier trained through an evolutionary algorithm. In: Tabii, Y., Lazaar, M., Al Achhab, M., Enneya, N. (eds.) BDCA 2018. CCIS, vol. 872, pp. 273–288. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96292-4_22
8. Ursani, Z., Corne, D.W.: A hierarchical set-partitioning nonlinear discriminant classifier trained by an evolutionary algorithm. In: 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD 2018), 26–28 May 2018, Chengdu, China. IEEE (2018)
9. Opitz, D., Maclin, R.: Popular ensemble methods: an empirical study. J. Artif. Intell. Res. **11** (1999), 169–198 (1999)
10. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016, in preparation). http://www.deeplearningbook.org, https://github.com/janishar/mit-deep-learning-book-pdf. Accessed 20 Mar 2018
11. Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H.G., Ogata, T.: Audio-visual speech recognition using deep learning. Appl. Intell. **42**, 722–737 (2015)
12. Chen, Y.C., Wang, J.S.: A Hammerstein-Wiener recurrent neural network with frequency-domain eigensystem realization algorithm for unknown system identification. J. Univ. Comput. Sci. **15**(13), 2547–2565 (2009)
13. Fisher, R.A.: The utilization of multiple measurements in taxonomic problems. Ann. Eugenics **7**, 179–188 (1936)
14. Raymer, M.L., Doom, T.E., Kuhn, L.A., Punch, W.F.: Knowledge discovery in medical and biological datasets using a hybrid Bayes classifier/evolutionary algorithm. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **33**(5), 802–813 (2003)
15. Bertsimas, D., Dunn, J., Pawlowski, C., Zhuo, Y.D.: Robust classification. INFORMS J. Optim. **1**(1), 2–34 (2019). https://doi.org/10.1287/ijoo.2018.0001
16. Bertsimas, D., Dunn, J.: Optimal classification trees. Mach. Learn. **2017**(106), 1039–1082 (2017)
17. Abellán, J., Mantas, C.J., Castellano, J.G., Moral-García, S.: Increasing diversity in random forest learning algorithm via imprecise probabilities. Expert Syst. Appl. **97**, 228–243 (2018)