Rajeev Kumar; Peter I Rockett

Decomposition of high dimensional pattern spaces for hierarchical classification

Kybernetika, Vol. 34 (1998), No. 4, [435]--442

Persistent URL: http://dml.cz/dmlcz/135228

KYBERNETIKA — VOLUME 34 (1998), NUMBER 4, PAGES 435 - 442

DECOMPOSITION OF HIGH DIMENSIONAL PATTERN SPACES FOR HIERARCHICAL CLASSIFICATION

RAJEEV KUMAR AND PETER ROCKETT

In this paper we present a novel approach to decomposing high dimensional spaces using a multiobjective genetic algorithm for identifying (near-)optimal subspaces for hierarchical classification. This strategy of pre-processing the data and explicitly optimising the partitions for subsequent mapping onto a hierarchical classifier is found to both reduce the learning complexity and the classification time with no degradation in overall classification error rate. Results of partitioning pattern spaces are presented and compared with various algorithms.

1. INTRODUCTION

Complex pattern recognition problems of high dimensionality are best addressed through a 'divide-and-conquer' approach rather than monolithically. A divide-and-conquer strategy decomposes a pattern space into a series of sub-spaces and assigns a set of function approximators to each subspace such that each module learns to specialise in a subdomain. This strategy however is only beneficial if a sensible method of partitioning the pattern space exists. One solution to this in a neural environment is competitive learning, i. e., decomposition-through-competition where decomposition and learning phases are combined. We argue that separating the task of decomposition from the regime of modular learning simplifies the overall architecture and this strategy of data pre-processing before its submission to a classifier considerably reduces the learning complexity.

In terms of its complexity, a partitioning problem is NP-complete and genetic algorithms (GAs) have been shown to be effective for exploring NP-complete search spaces relative to exhaustive search. GAs yield near-optimal solutions rather than an exact solution but have the advantage of not needing prior knowledge of the pattern space; the number of partitions that emerge from genetic search is guided solely by the optimisation criteria and is not dictated by user-defined parameters. Previous work on GA partitioning has optimised only a single, ad hoc objective and the eventual solutions have been strongly influenced by the linear coefficients used to combine different objectives into a single scalar fitness value. On the other hand, rank-based multiobjective genetic algorithms perform optimisation on a vector space of objectives and are able to explore the NP-complete search space for a set of

equally viable partitions of the space. For a detailed review of multiobjective genetic algorithms and partitioning, see [4].

In this work we address partitioning of the pattern space into a set of hyperspheres in a generic manner using a rank-based multiobjective genetic algorithm as a pre-processor to subspace learning for mapping onto a hierarchical neural classifier. Clusters are generated on the basis of 'fitness for purpose' – that is, they are explicitly optimised for their subsequent mapping onto the hierarchical classifier – rather than emerging as some implicit property of the clustering algorithm.

2. RATIONALE

Conceptually, our approach has strong links to recursive feature space partitioning algorithms using hyperplanes parallel to feature axes [1, 2]. The rationale for such hierarchical partitioning has been discussed by Kanal [3]. Although partitioning using a set of hyperplanes is simple, the method effectively gives rise to a decision tree classifier and has the drawback that it can be brittle: a wrong decision at a higher level of the tree usually leads to an unrecoverable error. In this work, we aim to partition feature spaces into subspaces which are each mapped onto a hierarchical neural network for efficient problem solving. If we consider feature-partitioning as a mapping p from an N dimensional feature space to j subspaces of dimensionality n_j:

$$p : R^N \rightarrow \bigcup_j R^{n_j}, \quad n_j \leq N \quad \text{subject to} \quad \text{Min/Max}\, Obj\_f_j(\boldsymbol{X})$$

then this formulation is an N dimensional function decomposition into many n_j-dimensional sub-functions subject to meeting certain criteria, $Obj\_f_i(\boldsymbol{X})$. Since n_j represents (hopefully) a less complex domain, a classifier can approximate such a sub-domain with less effort and one of the measures of complexity we employ is the local intrinsic dimensionality within a hyperspherical partition.

3. OBJECTIVES FOR SUBSPACE LEARNING

We have identified a set of seven distinct objectives for partitioning and optimising learning effort:

- *Minimise the number of hyperspheres* aims to exploit the modularity, but it should be based on minimising the *overall* training effort. Alternatively, this objective can be withdrawn and the number of partitioning hyperspheres can be specified in advance based on some prior knowledge of the problem domain.

- *Minimise the learning complexity* which, in a feedforward network, is approximately of the order of $O(N^3)$, where N is the number of weights in the network. For a given pattern space, the number of inputs and outputs remain fixed therefore the number of weights is proportional to the number of hidden units. It has been shown that the *effective* number of hidden units is (approximately) equal to the intrinsic dimensionality. Thus our objective is to minimise the sum of the cubes of the *intrinsic dimensionality* of the subspaces.

- *Maximise the regularity of the decision surface aims* at maximising the classification accuracy. The nearest neighbour classification error is used to indicate how well the partitions preserve the structure of the pattern space as a separability measure. This objective maximises the correct classification probability of a k-NN classifier within the partitions where we assume that each hypersphere is an 'independent' event in the statistical sense and we thus multiply their individual probabilities to combine the k-NN results.

- *Maximise the fraction of included patterns of each class* aims to include within all the partitions as many training patterns as possible from each class. Hopefully, outliers within the pattern space can be excluded because the objective does not aim to include all patterns.

- *Minimise the maximum fraction of included patterns in a single hypersphere* aims to distribute the included patterns over as many partitions as possible. It discourages inclusion of all the patterns in only one or a few partitions.

- *Minimise the overlap of partitions* aims to avoid repetition of learning effort on similar sets of patterns in different modules but allows some overlap of hyperspheres to prevent the formation of a 'no-man's land' between the partitions. For simplicity, we have used a simple measure of 'overlap' where we aim to 'push' apart two hyperspheres with a rudimentary coulombic-type repulsion model.

- *Minimise the surface area* attempts to produce compact solutions. The surface-content of a hypersphere is normalised by the number of patterns included within it so that the objective is biased towards compact solutions. This is effectively a data density measure.

All seven elements in the objective vector are distinct and competing as well as complementary to each other and ensure a fair distribution of potential solutions. Rather than using an *ad hoc* linear combination of these seven objectives we have employed the notion of Pareto optimality in which the superiority of one solution over another is measured in terms of 'dominance' resulting in an Pareto-optimal set which lies on a surface in the objective 7-space. At convergence, no single one of these objectives can be improved without degrading at least one of the others leading to a set of equivalent solutions. The implementation details of multiobjective genetic optimisation of subspaces are given in [4].

4. RESULTS

We have applied our partitioning strategy to a range of synthetic problems as well as a real classification problem. Firstly, we partitioned synthetic data from two, 3-dimensional gaussian 'blobs' embedded in a 6-space and within this we have examined two cases: one where the two blobs are just separated and the other where they overlap. Each class contained 100 examples. For the just-separated data a large number of equivalent solutions evolved, most of which comprised two clusters

of three *intrinsic* dimensions, each containing only data from the separate classes although some solutions contained exemplars from *both* classes. From the point of view of the GA, all (near-)optimal solutions are equivalent but some may be more desirable in practice. For the overlapped gaussian blobs, the GA produced partitions of intrinsic dimensionality of 3 or 4 and containing 5 – 10 % data from the other class, both of which would be expected for this dataset. Positioning two hyperspheres on the (known) gaussian centres and carrying-out an exhaustive search for the two 'best' hypersphere radii produced partitions which were comparable to the typical GA results indicating that the GA was indeed finding close-to- optimal clusters for both cases.

We have also considered the partitioning of a four-class synthetic problem in twelve variables; here each gaussian blob was of three (mutually exclusive) dimensions and just separated from the others. Again, we generated 100 random data points from each class. Most of the family of equivalent solutions produced comprised four clusters of three intrinsic dimensions, each containing around 100 data points. A number of equivalent solutions, however, contained seven or eight dimensional hyperspheres

Finally, we have partitioned a benchmark problem in land-use classification of multispectral satellite image data of thirty-six dimensions[1]. The dataset description and classification results for various algorithms are given in Taylor et al [5] where the best classification accuracy was obtained with a k-NN classifier. For simplicity, we reduced the original six-classes to a two-class problem (the cotton crop *versus* all others) and randomly sub-sampled the original 6435 data points to give a training set of 500. To investigate the behaviour of our partitioning algorithm in greater detail we fixed the number of clusters to 2, 4 and 6 in respective clustering runs; from viewing the data with standard ordination techniques it seemed that two clusters would be too few and six probably too many. Results for two hyperspheres produced some partitions which contained only members of one class and a roughly 50:50 split in the other partition. Most solutions, however, included a hypersphere containing around ten members of the other class. This latter situation is an unattractive partitioning since within one of the hyperspheres, one class has a very small prior which would lead to difficulties in reliably training a neural network. Partitions based on four clusters produced mostly hyperspheres of a single class together with hyperspheres of roughly 50:50 membership. Six partitions produced very similar results to the four partition case except that the aggregate overlap measure was increased and a few of the hyperspheres were degenerate in that they largely or wholly overlapped other hyperspheres.

We obtained classification results on the land use data based on the 500 training set examples and using the remaining 5935 examples as test data. Taking the whole, unpartitioned dataset produced a 3-NN error rate of 1.11 % and a 5-NN error rate of 1.23 %, whereas training a single feedforward neural network gave error rates of 1.23 % to 1.48 % dependent on architecture and the (random) initialisation.

[1] Available from the UCI Machine Learning repository at
http://www.ics.uci.edu/AI/ML/MLDBRepository.html or ELENA databases at
http://www.dice.ucl.ac.be/neural-nets/ELENA/ELENA.html.

The slightly poorer performance of MLPs relative to nearest neighbour classifiers is probably to be expected on such a dense data set. The misclassification rates for various methods are summarised in Figure 2. These observations are identical to those reported by Taylor et al [5] that k-NN is the best for this land-use data and so we have used 3-NN classification as a benchmark for our partitioning approach throughout this paper.

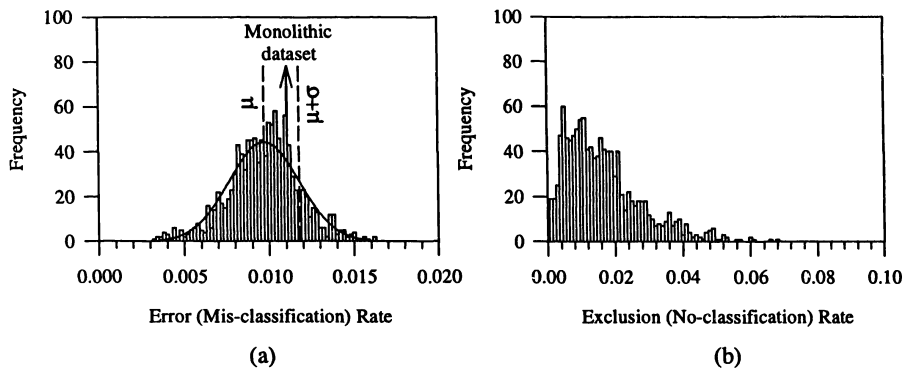

Fig. 1. (a) Mis-classification rate of unseen data (5935 patterns) computed with 3-NN classifier using 500 patterns as the training data. The histogram represents the error-rates of the partitioned (6 clusters each) solutions while the upward arrow indicates the error of the monolithic data. (b) The distribution of excluded patterns from the obtained solution.

Nearest neighbour error rates for the two- and four-cluster partitioned land use data were essentially identical to the results from treating the dataset as a monolithic block, and depending on the particular partitioning chosen, some were slightly better but none was worse. Mis-classification rates of six-cluster solutions are shown in Figure 1 (a) where the majority were slightly better and a few were worse. Assuming the results are normally distributed, the 3-NN error rate is within one standard deviation of the mean and thus we view any apparent improvement in error rate as not statistically significant. The fact that the classification accuracy was not reduced is highly significant; although the error rate has not changed *the computational effort for both training and recall is reduced by a large margin.*

Obviously the objective was not to include all training data within the clusters so some data were excluded from the solutions and these are potentially outliers. The distribution of the fraction of excluded points is shown in Figure 1 (b). Outliers do not tend to greatly degrade the performance of a k-NN classifier but they can have a serious effect in the case of a feedforward network. Hence performance improvement in the absence of outliers is not particularly marked with a k-NN classifier (Figure 1 (a)), but we believe it would be with the decision boundaries formed with feedforward networks – this is discussed further in subsequent paragraphs.

In an attempt to compare the genetic algorithm partitioning results with those of traditional clustering algorithms, we generated clusters in the range of [2...6] using the K-means clustering algorithm. For a fair comparison of the results, we generated the cluster centres with the same training data, and calculated the 3-NN error rates

for each cluster. The K-means clustering followed by a 3-NN classification within the bounding hyperspheres centred on each cluster gave error rates in the range of 1.21% to 1.71% dependent on the number of clusters and initial cluster centres. The traditional K-means clustering produced error rates of 1.35% to 2.48%. The error-rates of various models and algorithms are summarised in Figure 2(a).

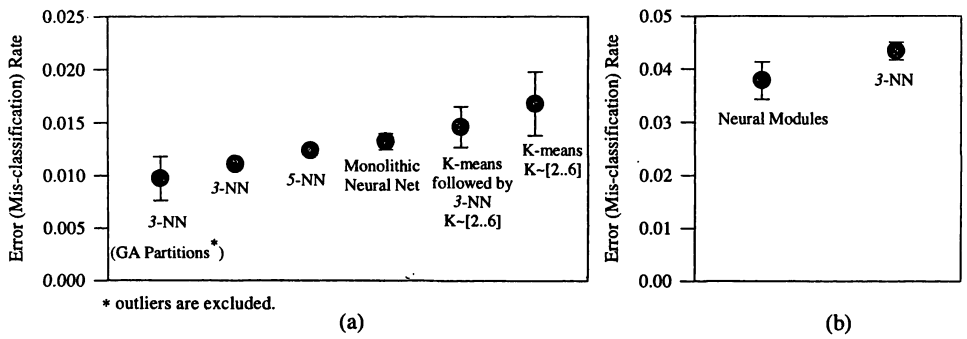

Fig. 2. A summary of mis-classification rates of unseen data (5935 patterns) using 500 training patterns (a) various models/algorithms, and (b) trivial partitions.

Looking at the composition of clusters across all the optimal solutions, we observed that only one or two clusters in each solution need post-partitioning classification. This is exactly what we are aiming for in this strategy since the localised decision surfaces should possess considerably reduced complexity over the global decision surface. We treated such 'split-class' partitions with both 3-NN and neural modules and measured the performance on the test data. The performance of the neural-modules was better than that of 3-NN classifier (Figure 2(b)) however the performance improvement is small and may be problem dependent. Nonetheless this is in keeping with the notion that on sparse datasets – and within a cluster the dataset is comparatively sparse – a properly trained neural network may be better able to generalise across the limited data available. This makes hierarchical neural classifiers a natural companion for the present partitioning approach. From the standpoint of the *time* required for classification, typically three-out-of-four or four-out-of-six partitions contained only a single class and so once inclusion within a hypersphere was established, labelling an unknown datum was trivial. Even for hyperspheres with a roughly equal split in the numbers of included classes, classification of an unknown point required far fewer nearest neighbour distance calculations than needed for classification based on the whole training set of five hundred. Such 'half-and- half' clusters typically contained 100-140 total patterns. Thus in nearest neighbour classification, our partitioned dataset gave error rates which were not degraded over a monolithic classifier, but the time to compute a label was reduced significantly.

5. DISCUSSION

From the above results, we observe that three main categories of cluster emerge: (i) all patterns within a hypersphere belong to a single class; (ii) a very few percent

of patterns belong to the other class; and (iii) both the classes are present in approximately equally numbers. The first category of clusters does not require any post-partitioning effort for classification since to label an unknown point it is sufficient to determine in which cluster it is included. In the third category of clusters, mapping on to a feedforward network is fairly straightforward since the roughly equal numbers of exemplars from each class together with the reduced size of the subset to be learned both simplify training. For the second broad category of cluster, a k-NN classifier could be employed to decide the final classification *within* a hypersphere with less computation than would be required for nearest neighbour classification on the whole training set although clearly, unless at least k members of the minority class are included, the classification effectively degenerates to the first category; we have observed a number of examples of clusters containing 200 members from one class and a single member from the other class. This present generic approach to partitioning as a pre-processor to the subsequent classifier is suited to a wide spectrum of complex problems, particularly where there is no prior knowledge of the pattern space which could guide clustering.

Most traditional clustering algorithms rely on some *similarity measure* and the resulting clusters depend directly on the judgement of what are and what are not nominally identical patterns. Our multiobjective GA approach avoids any such judgement of similarity and instead forms clusters on the basis of *fitness for purpose* - namely trying to simultaneously maximise a set of general properties we wish to emerge from a set of partitions. This property of genetic partitioning has been shown to exhibit superior results to those obtained from K-means clustering (Figure 2(a)). How the present partitioning technique handles outliers is the subject of further research.

6. CONCLUSIONS

In this paper we have presented a novel approach to partitioning pattern spaces using a multiobjective genetic algorithm for identifying (near-)optimal subspaces for hierarchical learning. The results of partitioning pattern spaces have been presented. This strategy of pre-processing the data and explicitly optimising the partitions for subsequent mapping on to a hierarchical classifier is found to both reduce the learning complexity and classification time for no statistically-significant change in overall classification error rate. Classification performance of various algorithms have been compared and it is argued that the neural-modules can be superior for learning the localised decision surfaces of such partitions as well as offering better generalisation than both a k-NN classifier and a monolithic neural network.

(Received December 18, 1997.)

REFERENCES

[1] J. H. Friedman: A recursive partitioning decision role for nonparametric classification. IEEE Trans. Comput. *26* (1997), 4, 404–408.

[2] E. G. Henrichon and K. S. Fu: A nonparametric partitioning procedure for pattern classification. IEEE Trans. Comput. *18* (1969), 7, 614–624.

[3] L. N. Kanal: Problem–solving models and search strategies for pattern recognition, IEEE Trans. Pattern Analysis Machine Intelligence *1* (1979), 2, 193–201.
[4] R. Kumar: Feature Selection, Representation and Classification in Vision. Ph.D. Thesis, Dept. Electronic and Electrical Engineering, University of Sheffield, 1997.
[5] C. C. Taylor et al: Dataset descriptions and results. In: Machine Learning, Neural and Statistical Classification (D. Michie, D. J. Spiegelhalter and C. C. Taylor, eds.), Ellis Horwood, London 1994, pp. 131–174.

Rajeev Kumar, Department of Computer Science & Information System, and Centre for Robotics & Intelligent System, Birla Institute of Technology & Science, Pilani. India. This work was done while he was at Department of Electronics & Electrical Engineering, University of Sheffield. England.
e–mail: rajeevk@bits.soft.net

Peter Rockett, Department of Electronics & Electrical Engineering, University of Sheffield. England.
e–mail: p.rockett@sheffield.ac.uk