Comparison of Two Classification Methodologies on a Real-World Biomedical Problem

Ray Somorjai[1], Arunas Janeliunas[2], Richard Baumgartner[1], and Sarunas Raudys[2]

[1] Institute for Biodiagnostics, NRCC
435 Ellice Avenue, Winnipeg, MB, Canada, R3B 1Y6
{ray.somorjai,richard.baumgartner}@nrc.ca
[2] Department of Mathematics and Informatics, Vilnius University
Naugarduko 24, LT, 2006 Vilnius, Lithuania
raudys@ktl.mii.lt
arunas.janeliunas@verslas.com

Abstract. We compare two diverse classification strategies on real-life biomedical data. One is based on a genetic algorithm-driven feature extraction method, combined with data fusion and the use of a simple, single classifier, such as linear discriminant analysis. The other exploits a single layer perceptron-based, data-driven evolution of the optimal classifier, and data fusion. We discuss the intricate interplay between dataset size, the number of features, and classifier complexity, and suggest different techniques to handle such problems.

1 Introduction

Many modern pattern classification and data mining problems are characterized by hundreds or thousands of attributes and huge amounts of data records. However, for most spectroscopy-based biomedical classification problems, although the number of attributes is large, data scarcity is the rule rather than the exception. Hence, relations between classifier complexity, feature space dimensionality and sample set size continue to be among the major research topics in pattern classification and data analysis.

Traditionally, complexity/ feature space dimensionality/sample size interrelations are tackled by first reducing the number of features, using some feature extraction/selection method [12]. An alternative approach is to adjust the classification rule to the number of training samples and feature space dimensionality.

A third way employs multiple classification systems (MCSs). In using MCS, the designer divides the attributes into non-intersecting or intersecting subsets and uses each subset of attributes to design a corresponding simple classifier ("expert"). Then the individual decisions of the experts are combined to arrive at the final decision. In each separate procedure, considerably fewer features may be required. In a modification of this approach, the designer divides the training records into separate non-intersecting or intersecting subsets and uses each subset to design a simple expert

T. Caelli et al. (Eds.): SSPR&SPR 2002, LNCS 2396, pp. 433-441, 2002.

classification rule instead of a complex one. This latter approach does need large sample sizes. In the last decade, the MCS approach has received considerable attention in the pattern recognition literature [1].

The existence of many similar approaches to solve the complexity/sample size/ dimensionality problem necessitates reviewing "the fundamental problems arising from this challenge" [2]. Notwithstanding theoretical attempts and advances, truly decisive tests of practical relevance can only be arrived at by experimental comparisons of the successes of different approaches on real-world problems. In the present paper, we conduct such comparisons of two strategies: MCS/classifier complexity regularization with a 3-stage feature-extraction-based statistical classification strategy (SCS) [9,10]. SCS was devised specifically for the classification of biomedical spectra. (The SCS's third stage is closely related conceptually to MCS.) The particular real-world example we use is a two-class biomedical data classification problem of typical difficulty. The observation vectors to be classified are magnetic resonance (MR) spectra of biofluids obtained from normal subjects and cancer patients. The dataset consists of 140 samples with 300 spectral features (the intensities at 300 frequencies). There are 71 spectra (31 healthy, 40 cancerous) in the training set and 69 (30 healthy, 39 cancerous) in the validation set. MR spectra of the 140 samples were acquired on a Bruker 360 MHz spectrometer. The MR magnitude spectra were preprocessed by normalizing each spectrum to unit spectral area. These are the data analyzed by both strategies.

2 A Feature-Selection-Based 3-Stage Statistical Classification Strategy (SCS)

The first stage of the SCS is feature selection. For MR magnitude spectra, the original N features are the intensity values at the different spectral frequencies. The feature selector algorithm we have used is an optimal region selector (ORS) [8]. ORS searches for spectral regions (frequency intervals) that are maximally discriminatory. ORS is guided by a genetic algorithm (GA), explicitly optimized for preprocessing spectra. GA is particularly appropriate for spectra, since the latter are naturally representable as "chromosomes", vectors of length N, with 1s indicating the presence, 0s the absence of features. The GA's input is M, the maximum number of features i.e., distinct spectral subregions required, the type of feature space-reducing operation/transformation (typically averaging) to be carried out, the population size, the number of generations and a random seed. The operations comprise the standard GA options: mutation and crossover. To ensure robust classification, the number of features M is typically kept much smaller than the sample size,. ORS begins searching the entire feature space, i.e. the complete spectrum. The output is the set of (averaged) spectral regions that optimally separate the classes.

For a limited number N of original features, exhaustive search (ES) for the best subset(s) is feasible. For larger N, we developed a dynamic programming (DP) based algorithm [8] that often produces near-optimal solutions, in feasible computer times.

Once M << N good features have been found, the second stage, a *crossvalidated classifier development* follows, with appropriately selected *training*, *test* and

validation sets. We have developed an approach (RBS, Robust "BootStrap") that was inspired by the conventional nonparametric bootstrap [15]. RBS proceeds by randomly selecting approximately half the spectra from each class and using these to *train* a classifier, usually linear discriminant analysis (LDA). The resulting classifier is then used to *validate* the remaining half of the spectra. This process is repeated B times (*with replacement*), and every time the optimized LDA coefficients are saved. (B is typically 500-1000.) The *weighted average* of these B sets of coefficients produces the final, single *W-weighted classifier*. The weight for the *m*th set is $W_m = \kappa_m C_m^{1/2}$, m = 1,…,B, where $0 \le C_m \le 1$ is the *crispness* (for two classes, the fraction of samples with class probability ≥ 0.75), and $\sim 0 \le \kappa_m \le 1$ is Cohen's [16] chance-corrected *measure of agreement*, $\kappa_m = 1$ signifying perfect classification. The B weight values W_m are those obtained for the less optimistic bootstrap *test sets*. Classifier outcome is reported as a *class probability*.

For difficult classification problems the third stage is activated. At this stage, the outcomes of several classifiers are combined into an overall classifier via classifier fusion methods [13,14]. We use Wolpert's Stacked Generalizer [17] (WSG) for classifier combination. For the ultimate classifier to be developed, the *input features* for WSG are the *output* class *probabilities* obtained by the individual classifiers. For 2-class problems (since the *two* class probabilities are not independent, $p_1 + p_2 = 1$), the number of such features is *one* probability per sample (spectrum). The overall classification quality of the fused classifier is generally higher than that of the individual classifiers. In particular, the *crispness* of this final classifier is invariably greater. This is important in a clinical setting, because greater class assignment certainty means that fewer patients will have to be re-examined.

3 Controlling Classifier Complexity by Training a Single Layer Perceptron

There are many pattern classification strategies. They may differ conceptually, in the assumptions used to establish the design procedure, in the way the parameters of the classifier are estimated, and in the complexity of the decision boundary. When the design set is small, one promising approach is to use a linear classifier obtained while training a single layer perceptron (SLP). An important characteristic of an SLP-based classifier is that while training the perceptron, a number of standard pattern classification algorithms of differing complexity can be obtained by simply changing certain conditions [3,4]. Moreover, prior to training the perceptron, one can use sample estimates of statistical parameters of the training data to perform data transformations (rotation and scaling) such that various statistical models (i.e., prior information about the problem) may be incorporated into the perceptron design [4].

We followed the procedure described in [4] and prior to training the perceptron:

- The data centre $\hat{\boldsymbol{M}} = ½ (\hat{\boldsymbol{M}}_1 + \hat{\boldsymbol{M}}_2)$ was zeroed, and all single features were scaled to unit variance by their sample standard deviations s_i;
- An estimate $\mathbf{S}_e$ of the pooled covariance matrix (CM) of the training data was constructed, followed by a singular-value-decomposition of $\mathbf{S}_e$, to linearly

transform the data to $\boldsymbol{Y} = \mathbf{F}(\boldsymbol{X} - \hat{\boldsymbol{M}})$, where $\mathbf{F}= \mathbf{\Lambda}^{-1/2}\ \mathbf{\Phi}^{T}$, and $\mathbf{\Lambda}$, $\mathbf{\Phi}^{T}$ are eigenvalues and eigenvectors of the $\mathbf{S}_e$;

- Training started from zero initial weights;
- The standard sum of squares cost function and total gradient training (batch mode) was used.

To be able to work in the original 300-dimensional spectral feature space, given the very small training sets, it was necessary to use some prior information about the data. We assumed that all mutual correlations among all features are equal, and depend only on the noise variance. Hence, the correlation matrix $\mathbf{S}_e$ will be characterized by a single parameter, the correlation coefficient ρ.

This model, describing the dependence among the data features, is called the *additive noise model* Accordingly, after subtracting the mean vector $\hat{\boldsymbol{M}}$ and scaling to unity the variances of all features, we calculated the correlation matrix and used the average value of the correlation coefficient to obtain $\mathbf{S}_e$. Then we used this matrix to perform the singular value decomposition and to rotate the input feature vectors. In the new, transformed feature space, the main dependency between the components of the feature vector is already taken into account and the training process is faster. Optimal stopping of the iterative training process, control of target values, and the addition of an antiregularization term to the cost function can help balancing the complexity of the classification rule and the training set size.

4 Multiple Classifier Systems

In order to design an MCS, we partitioned the MR magnitude spectral features into 12 non-overlapping subsets. Then on each of these subsets, we constructed a simple classification rule by training a single-layer perceptron with the exponential threshold function $f(x) = 1/(1 + e^{-x})$. The outputs (i.e., the values of f(x)) of these single expert classifiers actually served as new input features for the "governor" SLP training.

For a small design set, it is very difficult to determine the optimal number of SLP training iterations. In such cases, one must use the same training set both to validate classifier performance and to define a stopping point for SLP training. This method of classification performance estimate is called the resubstitution error estimate.

Another approach is to create the independent validation data from random noise vectors, by augmenting the training set with them. One usually injects "white" noise vectors from a Gaussian distribution $N(0, \lambda \boldsymbol{I})$, where $\boldsymbol{I}$ is the identity matrix and λ is some scalar. However, further improvement is obtainable for high-dimensional problems, by adding instead *k-NN*-directed "colored" noise [6]. We used this technique to produce the validation dataset for training the expert classifiers. The outputs of these expert classifiers for the validation data provided the validation dataset for governor-SLP training.

We performed several experiments with different expert-SLP training techniques in order to compare the resulting classification performances. In our first attempt to design expert-SLP classifiers, we simply followed the recommendations in [4]. We scaled and rotated the original training data vectors and then trained the SLP

classifier, starting from zero weights. No other assumption about the data was used. We will refer to this set of expert classifiers as the "simple" SLP experts.

During training, the single-layer perceptron tends to adapt to the specific training dataset (often called overtraining). When we use the resubstitution error to estimate classifier performance, each expert classifier "boasts" to the fusion rule, i.e., overestimates its own accuracy. Thus, if the same data are used to train both the classifiers and the combiner, the outcomes of the classifiers to the training data vectors create an optimistically biased training data for the fusion rule. We call this "boasting bias". Therefore, our next attempt was to improve the classification performance of our MCS by correcting the boasting bias of the simple SLP experts.

Assuming that the simple SLP experts are similar to Fisher discriminant functions (FDF), we have applied FDF-type theoretical corrections to the outputs of the SLP experts. Using the mean and variance values of the outputs of FDF classifiers given in [7], we define the following boasting-bias-correcting (BBC) transformation of the outputs of expert classifiers:

$$\tilde{O}_i = N/(N-p_i)\, O_i + (-1)^j N p_i(\delta_i^2+4)/2(N-p_i)^2 \quad (1)$$

where O_i is the original output of the i-th classifier, N is the total number of training vectors, p_i is the dimensionality of the training data for the i-th classifier, δ_i^2 is the squared Mahalonobis distance between the two classes for the i-th classifier, and j is the class number. These corrections change the experts' means and variances. Thus, we produced a new, corrected training dataset for the governor-SLP, anticipating that it will help us obtain a better classifier fusion rule.

However, the simple SLP experts are obviously not FDF classifiers and such FDF-oriented BBC may not be appropriate. A more suitable BBC rule is the one that most affects the common distribution parameters of the expert classifier outputs. Hence, we tried a simpler BBC technique that affects only the means of the expert outputs, as shown in [7]:

$$\tilde{O}_i = O_i + (-1)^j 2p_i/(N-p_i), \quad (2)$$

This was the second corrected training dataset for the governor-SLP training.

The second group of SLP experts was designed using the additive noise model. We used the previous splitting of data features into 12 subsets, and then we trained the individual SLP classifiers using the complexity control techniques described in Section 3. We assumed equal mutual correlations among data features, hence introduced into the expert SLP training additional information about the data structure.

We also trained the governor-SLP using the additive noise model. The outputs (or corrected outputs) of the SLP experts for the training data were used as input features for governor-SLP training, and the outputs for the noise data were used to stop the governor-SLP training. However, the role of colored noise vectors in the training of expert classifiers can also be changed. If we assume that we can create as many noise vectors as we need, and that the *k-NN*-directed colored noise retains information about the data configuration, we can use it as the training data for the governor-SLP. Furthermore, we can use the real training data for a more reliable determination of the number of governor-SLP training iterations than is possible by the random noise vectors.

5 Comparison Experiments

For the experiments with both the SCS and the CCR & MCS, we partitioned the data into two subsets. There are 71 spectra (31 healthy, 40 cancerous) in Set 1, and 69 (30 healthy, 39 cancerous) in Set 2. We then performed two runs of experiments:

Run 1: Train on Set 1, validate on Set 2
Run 2: Train on Set 2, validate on Set 1.

5.1 Experiments with the SCS: Dynamic Programming (DP) & Exhaustive Search (ES) on the Original Attributes

Because the number of attributes is relatively small (300), we did not need to use the more sophisticated, genetic algorithm-driven optimal region selector preprocessor; instead, we applied the DP-based feature selection algorithm.

Run 1: We first applied DP to Set 1. Because this is a 2-class problem, we could use a classifier that is a robust equivalent of LDA (we employed least-trimmed-squares regression, 10% trimming), with leave-one-out (LOO) crossvalidation. For the attribute selection, we used an objective function F that simultaneously minimizes the squared classification error and maximizes the crispness. In the range 2-13 of requested number of attributes, the minimum F for Set 2, *the validation set*, was obtained by the 8 attributes 8, 15, 18, 29, 115, 124, 151, 265. Because DP is a suboptimal feature selector, and to avoid overfitting, we used ES to select the best 2-7 subsets of these 8 attributes. The best of these, comprising only 3 attributes (8, 124, 151) gave a misclassified percentage of 13.0% for Set 2. The crisp result was 5.3% (38 of 69, 55.1% of total). A total of (12+6) = 18 models were tested.

Run 2: From the original 300 attributes DP selected 30 (using 6 tries, i.e., models). From these 30, the best 5 chosen by ES were 26, 151, 164, 165 and 248 (8 models). The misclassification percentage for the switched validation set (Set 1) was 14.1%, the crisp result 14.3% (70 of 71, 98.6%). (6+8) = 14 models were tested.

Averaging the two runs yielded 13.6%; based only on the crisp assignments, the average was 9.8% (108 of 140, 77.1% of total).

5.2 Experiments with CCR & MCS

For each run, we designed two sets of SLP-experts: 12 simple SLP classifiers and 12 SLP classifiers with the additive noise model. We used *3-NN*-directed colored noise vectors as the independent validation data set. For each training vector we added 300 colored noise vectors with variance $\lambda = 0.5$.

The governor-SLP was trained both on the training data and on the noise data. The optimal number of training iterations was determined by using the real validation data vectors. When the noise data (or the real training data, when we used noise data for training) was used to stop the governor-SLP training, we have the non-optimally stopped governor-SLP.

In the table below, we present the percentages of misclassified validation data vectors. To produce the results, 6 different feature distributions were tested for the

SLP experts (3 using the information measure, 3 the correlation matrix). For each of these, 3 different values of the correlation coefficient were tried in the additive noise model. From this total, (3+3)*3 = 18 different versions, the best results are listed in the table.

Expert classifiers	Average % performance of expert classifiers			Real training data (Noise training data)					
				Optimally stopped governor-SLP			Non-optimally stopped governor-SLP		
	1st run	2nd run	Avg	1st run	2nd run	Avg	1st run	2nd run	Avg
Simple SLP	39.6	36.6	38.1	18.8 (27.5)	26.8 (22.5)	22.8 (25.0)	26.1 (29.0)	28.2 (26.8)	27.1 (27.9)
Correction Eq. (1)	-	-	-	20.3 (26.1)	28.2 (25.4)	24.2 (25.7)	27.5 (29.0)	28.2 (26.8)	27.9 (27.9)
Correction Eq. (2)	-	-	-	17.4 (21.7)	26.8 (22.5)	22.1 (22.1)	29.0 (31.9)	26.8 (26.8)	27.9 (29.3)
Additive noise model	34.3	0.4	37.3	14.5 (13.0)	22.5 (21.1)	18.5 (17.1)	18.8 (21.7)	28.2 (22.5)	23.5 (22.1)

6 Discussion and Concluding Remarks

For the first time, two comprehensive classification strategies, developed independently by the Vilnius and Winnipeg groups, were compared on a real-world dataset not commonly accessible to the machine intelligence community.

The differences between the results obtained by the two strategies are not significant statistically. Nevertheless, the two strategies arrived at these results by quite different routes, using somewhat different philosophies. Comparing individually the misclassified test samples produced by the two best classifiers (13.0% both for the SCS-based approach and for the MCS method, the latter for the additive noise model with noise training data and optimally stopped governor-SLP) shows that the two approaches misclassified the same number of, but not the same individual samples.

The focus of the SCS is selecting maximally discriminatory features, by various preprocessing approaches. Extensive experience with biomedical spectra (see references in [10]) indicates that when the number of appropriate features extracted from the spectra is ~1/5th – 1/10th of the number of spectra per class, even a simple linear classifier, such as LDA will be reliable, once properly crossvalidated. The third stage, classifier fusion, is invoked only if the outcome probabilities are low. This was not done for this study.

In contrast, the MCS approach starts with a combination of implicit feature selection and classifier fusion (governor-SLP). (Unlike in the SCS, where classifier fusion is via the output class probabilities that serve as input features for the ultimate classifier, MCS uses the outputs O_i of the SLP experts as inputs to the governor-SLP.) However, the overall strategy's most distinguishing aspect is the use of a data-driven

selection of the optimal classifier (CCR), which may range from the simplest (Euclidean distance classifier) to the most complex (support vector machines).

For the MSC approach, we tested several versions of the strategy (different subsets of experts, optimal stopping, etc). For the SCS, we selected the best attributes, based on the validation set's classification accuracy. Therefore, in both cases we adapted to the validation data, leading to optimistically biased results. Ideally, one would need a sufficiently large (hence statistically significant), completely independent validation set that was never used in the actual classifier development. This, at least for biomedical or genomics (microarray) applications, is unrealistic, and reliable methods for augmenting an originally sparse dataset (e.g., by noise injection) become particularly relevant.

Clearly, there is no best universal strategy or classifier! Hence, one cannot decide in advance what classification strategy and/or classifier to use. For each particular situation and dataset, comparative experimentation will be necessary. For real-life data, the designer will have to test several different models to decide which is best. This is an important consideration for the final assessment of the classifier(s). An essential requirement of success is that the classification strategy be comprehensive and sufficiently flexible to adapt to the peculiarities of the data. This is highlighted in this study: although the emphasis may be different, both strategies rely, explicitly or implicitly, both on feature selection and on classifier fusion. (Note that the explicit feature selection stage of the SCS, designed to create features that retain spectral identity, is driven by the biomedical imperative to understand the biochemical origin of the diseases.) However, the SCS and MCS place different emphasis on the components of the strategies, and carry them out in a different order. The major differences are SCS's reliance on feature selection vs. MCS's data-tuned classifier development.

We recommend having a toolbox of different classification strategies, and that the user experiment to select the most appropriate for the task. In the present context, a fusion of the best components of the two strategies used above seems promising. We shall report on these experiments in a future communication.

In general, to understand better the intricate interplay between dataset size, the number of features, and classifier complexity, it is highly desirable that several different classification experiments be performed on many different *types* of real-life datasets (beyond those archived in a few machine intelligence-oriented databases). Furthermore, both details of positive and negative results should be reported.

References

1. Kittler J., Roli F. (eds): Multiple Classifier Systems. Springer Lecture Notes in Computer Science, Springer Vol. 1857 (2000), Vol. 2096 (2001)
2. Ho T.K.: Data complexity analysis for classifier combination. In: Multiple Classifier Systems. J. Kittler and F. Roli (eds). Springer Lecture Notes in Computer Science, Springer Vol. 2096, (2001), 53-67
3. Raudys S.: Evolution and generalization of a single neuron. I. SLP as seven statistical classifiers. Neural Networks 11, 1998, 283–96

4. Raudys S.: Statistical and Neural Classifiers: An integrated approach to design. Springer, London, (2001) 312
5. Pivoriunas V.: The linear discriminant function for the identification of spectra. In: S Raudys (editor), Statistical Problems of Control 27, (1978), 71–90. Institute of Mathematics and Informatics, Vilnius (in Russian)
6. Skurichina M., Raudys S., Duin R.P.W.: K-nearest neighbours directed noise injection in multilayer perceptron training. IEEE Trans. On Neural Networks. 11(2) (2000), 504-511
7. Janeliūnas A.: Bias correction of linear classifiers in the classifier combination scheme. In: Proceedings of the 2nd International Conference on Neural Networks and Artificial Intelligence, BSUIR, Minsk, (2001) 91-98
8. Nikulin A., Dolenko B., Bezabeh T., Somorjai R.: Near-optimal region selection for feature space reduction: novel preprocessing methods for classifying MR spectra. NMR in Biomedicine, 11 (1998) 209-216
9. Mountford C., Somorjai R., Gluch L., Malycha P., Lean C., Russell P., Bilous M., Barraclough B., Gillett D., Himmelreich U., Dolenko B., Nikulin A., Smith I.: MRS on breast fine needle aspirate biopsy determines pathology, vascularization and nodal involvement. Br. J. Surg. 88 (2001) 1234-1240
10. Somorjai R.L., Dolenko B., Nikulin A., Nickerson P., Rush D., Shaw A., de Glogowski M., Rendell J., Deslauriers R. (2002) Distinguishing normal allografts from biopsy - proven rejections: application of a three - stage classification strategy to urine MR and IR spectra. Vibrational Spectroscopy 28: (1) 97-102
11. Zhilkin P., Somorjai R.: Application of several methods of classification fusion to magnetic resonance spectra. Connection Science 8(3) (1996) 427-442
12. Jain A.K., Duin R.P.W., Mao J.: Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 4-37
13. Somorjai R.L., Nikulin A.E., Pizzi N., Jackson D., Scarth G., Dolenko B., Gordon H., Russell P., Lean C.L., Delbridge L., Mountford C.E., Smith I.C.P.: Computerized consensus diagnosis: a classification strategy for the robust analysis of MR spectra. I. Application to 1H spectra of thyroid neoplasms. Magn. Reson. Med. 33 (1995) 257-263
14. Somorjai R.L., Dolenko B., Nikulin A.E., Pizzi N., Scarth G., Zhilkin P., Halliday W., Fewer J., Hill N., Ross I., West M., Smith I., Donnelly M., Kuesel A., Brière K.: Classification of 1H MR spectra of human brain biopsies: The influence of preprocessing and computerized consensus diagnosis on classification accuracy. J Magn Reson Imaging 6 (1996) 437-444
15. Efron B., Tibshirani R.: An Introduction to the Bootstrap. Monographs on Statistics and Applied Probability, Cox D., Hinkley D., Reid N., Rubin D. and Silverman B.W. (General Eds.) Vol. 57 Chapman & Hall, London (1993)
16. Cohen J.: Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. Psychological Bulletin 70 (1968) 213-220
17. Wolpert D.H.: Stacked generalization. Neural Networks 5 (1992) 241-259