Bagging and the Random Subspace Method for Redundant Feature Spaces

Marina Skurichina and Robert P.W.Duin

Pattern Recognition Group, Department of Applied Physics, Faculty of Applied Sciences, Delft University of Technology, P.O. Box 5046, 2600GA Delft, The Netherlands
{marina, duin}@ph.tn.tudelft.nl

Abstract. The performance of a single weak classifier can be improved by using combining techniques such as bagging, boosting and the random subspace method. When applying them to linear discriminant analysis, it appears that they are useful in different situations. Their performance is strongly affected by the choice of the base classifier and the training sample size. As well, their usefulness depends on the data distribution. In this paper, on the example of the pseudo Fisher linear classifier, we study the effect of the redundancy in the data feature set on the performance of the random subspace method and bagging.

1 Introduction

In many applications of discriminant analysis, data often consist of a large number of measurements (features) with a relatively small number of observations (objects). In these circumstances, it may be difficult to construct a good single classification rule. Usually, a classifier, constructed on small training sets is biased and has a large variance. Consequently, such a classifier may be weak, having a poor performance [1]. One way to improve a weak classifier is to stabilize its decision, for instance, by regularization [2] or noise injection [3]. Another popular approach is to use a combined decision of many weak classifiers instead of a single classifier. The examples of such combining techniques are bagging [4], boosting [5] and the Random Subspace Method (RSM) [6], which are originally designed for decision trees.

When applying bagging and the RSM to Linear Discriminant Analysis (LDA), it was established that these techniques are useful in different situations. Their performance is strongly affected by the choice of the base classifier and the training sample size [7]. As well, it was noticed that their relative performance differs for different data sets. It was demonstrated that the problem complexity (feature efficiency, the length of class boundary etc.) affects the performance of combining techniques [8]. In particular, it was shown that the RSM performs relatively better when a discrimination power is distributed evenly over many features.

In this paper we study the effect of the redundancy in the feature set on the performance of bagging and the RSM. When data have many completely redundant noise features or many data features are highly correlated (many features contain the same information), the intrinsic (true) data dimensionality is smaller than the dimensionality of the feature space where data objects are described. The intrinsic data dimensionality is one of the important characteristics of the data set that may influence the performance of classifiers. Consequently, the redundancy in the data feature set (and the intrinsic data dimensionality) may affect the performance of the combining techniques, as their performance depends on the training sample size referred to the data dimensionality. It is especially of interest for the RSM, where one constructs classifiers in random subspaces, because the number of informative features and the dimensionality of random subspaces should affect the performance of the RSM.

In order to study the effect of redundancy in the data feature set on the performance of bagging and the RSM (which are discussed in section 2), we consider two cases of the redundancy representation in the data feature set. In the first case, data have many completely redundant noise features. In the second case, the useful information is spread over many features while the data itself have a low intrinsic dimensionality. The used data sets representing a two-class problem are described in section 3. We perform our simulation study on the example of the Pseudo Fisher Linear Discriminant (PFLD) [9]. Using this classifier as a single classification rule is not recommended because it is weak for critical training sample sizes (when the number of training objects is comparable with the data dimensionality) (see Fig. 1). However, bagging and the RSM are just designed for weak classifiers. So the PFLD is very suitable to be used as a base classifier in these combining techniques when it is constructed on critical training sample sizes. Moreover, our previous study [7] has shown that bagging and the RSM are useful for the PFLD. The results of our simulation study of the effect of the redundancy in the data feature set on the performance of bagging and the RSM are discussed in section 4. Conclusions are summarized in section 5.

2 Bagging and the Random Subspace Method

In order to improve the performance of weak classifiers, a number of combining techniques can be used. Bagging and the RSM are two of them. They both modify the training data set by sampling either training objects (in bagging) or data features (in the RSM), build classifiers on these modified training sets and then combine them into a final decision. Usually the simple majority vote is used to get a final decision. However, the weighted majority vote used in boosting [5] is more preferable because it is more resistant to overtraining (when increasing the number B of combined classifiers) than other combining rules [10]. Therefore, we use the weighted majority vote in both studied combining techniques.

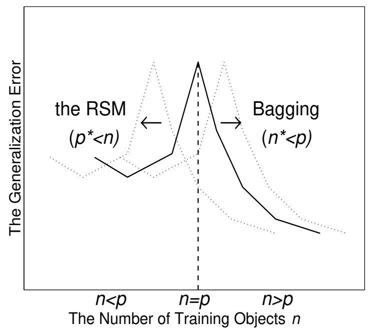

Fig. 1. The shifting effect of the *generalization error* (GE) of the PFLD for the RSM and bagging. In the RSM, the GE shifts with respect to the GE of the original classifier in the direction of the GE obtained on larger training sets. In bagging, the GE shifts with respect to the GE of the original classifier in the direction of the GE obtained on smaller training sets. (Here, n is the number of training objects and p is the data dimensionality.)

Bagging is proposed by Breiman [4] and based on bootstrapping [11] and aggregating concepts. By that, it incorporates benefits of both approaches. Bootstrapping is based on random sampling with replacement. Therefore, taking a bootstrap replicate $\boldsymbol{X}^b = (\boldsymbol{X}_1^b, \boldsymbol{X}_2^b, \ldots, \boldsymbol{X}_n^b)$ (the random selection with replacement) of the training set $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n)$, one can sometimes avoid or get less misleading training objects in the bootstrap training set. Consequently, a classifier constructed on such a training set may have a better performance. Aggregating actually means combining classifiers. Often a combined classifier gives better results than individual classifiers. Therefore, bagging might be helpful to build better classifiers on training sample sets with misleaders. In bagging, bootstrapping and aggregating techniques are implemented in the following way.

1. Repeat for b=1,2,...,B.
   a) Take a bootstrap replicate $\boldsymbol{X}^b$ of the training data set $\boldsymbol{X}$.
   b) Construct a classifier $C^b(\boldsymbol{x})$ (with a decision boundary $C^b(\boldsymbol{x}) = 0$) on $\boldsymbol{X}^b$.
   c) Compute combining weights $c_b = \frac{1}{2}\log\left(\frac{1-err_b}{err_b}\right)$, where $err_b = \frac{1}{n}\sum_{i=1}^{n} w_i^b \xi_i^b$ and $\xi_i^b = \begin{cases} 0, \text{ if } X_i \text{ is classified correctly} \\ 1, \text{ otherwise} \end{cases}$.
2. Combine classifiers $C^b(\boldsymbol{x})$, b=1,2,...,B, by the weighted majority vote with weights c_b to a final decision rule $\beta(\boldsymbol{x}) = \underset{y \in \{-1, 1\}}{argmax} \sum_b \delta_{\mathrm{sgn}(C^b(\boldsymbol{x})), y}$, where $\delta_{i,j} = \begin{cases} 1, i = j; \\ 0, i \neq j; \end{cases}$ is Kronecker symbol, $y \in \{-1, 1\}$ is a decision (class label) of the classifier.

Bootstrapping is most efficient when the training set is in order of the data dimensionality. By this, bagging is useful for linear classifiers constructed on critical training sample sizes, when they are unstable [12]. When bootstrapping the training set in bagging, in average only 1-1/e= 63.2% of the training objects is used in each bootstrap replicate. In this way, the bootstrap sample is comparable with a smaller training set. So, the bagged classifier will have similar characteristics as the classifier built on the smaller training set. *The generalization error of the bagged classifier is shifted* with reference to the generalization error of the original classifier in the direction of the generalization error obtained on a smaller training set (see Fig. 1). Thus, *the performance of bagging depends on the training sample size related to the data dimensionality and on the small sample size properties of the base classifier.* It implies that, when applied to LDA, *bagging is useful for classifiers with a non-decreasing learning curve* (the dependency of the classification error as a function on the training sample size) *constructed on critical training sample sizes* [12].

The *Random Subspace Method* is the combining technique proposed by Ho [6]. In the RSM, one also samples the training data. However, the sampling is performed in the feature space. Let each training object $\boldsymbol{X}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ (i=1,...,n) in the training sample set $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n)$ be a p-dimensional vector, described by p features. In the RSM, one randomly selects p*<p features from the p-dimensional data set $\boldsymbol{X}$. By this, one obtains the p*-dimensional random subspace of the original p-dimensional feature space. So the modified training set $\tilde{\boldsymbol{X}}^b = (\tilde{\boldsymbol{X}}_1^b, \tilde{\boldsymbol{X}}_2^b, \ldots, \tilde{\boldsymbol{X}}_n^b)$ consists of p*-dimensional training objects $\tilde{\boldsymbol{X}}_i^b = (x_{i1}^b, x_{i2}^b, \ldots, x_{ip*}^b)$ (i=1,...,n), where p* components x_{ij}^b (j=1,...,p*) are randomly selected from p components x_{ij} (j=1,...,p) of the training vector $\boldsymbol{X}_i$ (the selection is the same for each training vector). Then one constructs classifiers in the random subspaces $\tilde{\boldsymbol{X}}^b$ and aggregates them b in the final decision rule. Namely, the RSM is organized in the following way.

1. Repeat for b=1,2,...,B.
   a) Select the p^* -dimensional random subspace $\tilde{X}^b$ from the original p-dimensional feature space X .
   b) Construct a classifier $C^b(\boldsymbol{x})$ (with a decision boundary $C^b(\boldsymbol{x}) = 0$) in $\tilde{X}^b$.
   c) Compute combining weights $c_b = \frac{1}{2}\log\left(\frac{1 - err_b}{err_b}\right)$, where $err_b = \frac{1}{n}\sum_{i=1}^{n} w_i^b \xi_i^b$ and $\xi_i^b = \begin{cases} 0, \textit{ if } \ X_i \ \textit{is classified correctly} \\ 1, \textit{otherwise} \end{cases}$.
2. Combine classifiers $C^b(\boldsymbol{x})$, b=1,2,...,B, by the weighted majority vote with weights c_b to a final decision rule $\beta(\boldsymbol{x}) = \underset{y \in \{-1, 1\}}{argmax} \sum_b \delta_{\mathrm{sgn}(C^b(\boldsymbol{x})), y}$, where $\delta_{i, j} = \begin{cases} 1, \ i = j; \\ 0, \ i \neq j; \end{cases}$ is Kronecker symbol, $y \in \{-1, 1\}$ is a decision (class label) of the classifier.

The RSM may benefit from both, using random subspaces for constructing the classifiers and aggregating the classifiers. In the case, when the number of training objects is relatively small as compared with the data dimensionality, by constructing classifiers in random subspaces one may solve the small sample size problem, because the training sample size relatively increases in random subspaces. When data have many redundant features, one may obtain better classifiers in random subspaces than in the original feature space. The combined decision of such classifiers may be superior to a single classifier constructed on the original training set in the complete feature space.

The performance of the RSM is also affected by the training sample size and the small sample size properties of the base classier [7]. The subspace dimensionality is smaller than in the original feature space while the number of training objects remains the same. By this, the relative training sample size increases. Similar to bagging, *in the RSM, the final classifier will have a shifting effect of the generalization error* with respect to the generalization error of the original classifier (see Fig. 1). However, this shift will be in the opposite direction: in the direction of the generalization error obtained on larger training sample sizes. So *the RSM is useful for classifiers having a decreasing learning curve constructed on small and critical training sample sizes*.

Thus, bagging and the RSM may be useful for linear classifiers constructed on critical training sample sizes. However, bagging is beneficial for linear classifiers with a non-decreasing learning curve, while the RSM is useful for linear classifiers having a decreasing learning curve. The PFLD is weak when it is constructed on critical training sample sizes, having a peak of the generalization error at point n=p (see Fig. 1). By this, the learning curve of the PFLD increases for training sample sizes n<p, and decreases for n>p. Therefore, both, bagging and the RSM, may improve the performance of the single PFLD constructed on critical training sample sizes.

3 Data

In our experimental investigations we considered one artificial and five real data sets representing a two-class problem, which we have modified in order to get data sets having a high redundancy in the data feature space.

The artificial data set used to obtain data with many redundant features is the *2-dimensional correlated Gaussian data* set constituted by two classes with equal covariance matrices. Each class consists of 500 vectors. The mean of the first class is zero for both features. The mean of the second class is equal to 3 for both features. The common covariance matrix is a diagonal matrix with a variance of 40 for the second feature and unit variance for the first feature. This data set is rotated by using a rotation matrix $\boldsymbol{R} = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$.

The real data sets, used to obtain highly redundant data sets, are taken from the UCI Repository [14]. They are the 34-dimensional *ionosphere* data set, the 8-dimensional *pima-diabetes* data set, the 60-dimensional *sonar* data set, the 30-dimensional *wdbc* data set and the 24-dimensional *german* data set.

In order to make these data sets highly redundant in the feature space, to each of them we added r completely redundant noise features distributed normally $N(0,10^{-4})$. Thus, we have obtained $p+r$-dimensional data sets with many completely redundant noise features (one way of the redundancy representation in the data feature set). Here, p is the dimensionality of the original data set. In order to obtain data sets where a useful information is spread over all $p+r$ features (other way of the redundancy representation in the feature set), we have rotated the data enriched by noise features. The rotation was performed in all $p+r$ dimensions by using a Hadamard matrix [15]. By this, we have obtained the data sets where the classification ability (discrimination power) is spread over all features.

Training sets are chosen randomly from a total data set. The remaining data are used for testing. All experiments are repeated 50 times on independent training sets. So all figures show the averaged results over 50 repetitions. The standard deviations of the mean generalization errors for the single and combined PFLD's are around 0.01 for each data set.

In bagging and the RSM, ensembles consist of 100 classifiers combined by the weighted majority vote in the final decision. The size p^* of random subspaces used in the RSM is 10.

4 The Effect of the Redundancy in the Data Feature Space

In order to study the effect of redundancy in the data feature set on the performance of bagging and the RSM, we have modified several data sets by artificially increasing their redundancy in the data feature set (see previous section). In order to investigate whether redundancy representation in the data feature space affects the performance of the combining techniques, we modify the original data sets in two different ways: 1) just adding r completely redundant noise features to the original p-dimensional data set (by this, a discrimination power is condensed in the subspace described by the original p-dimensional data set), 2) additionally rotating the data set in all $p+r$ dimensional space after injecting r completely redundant noise features to the original data set (by this, a discrimination power is spread over all $p+r$ features).

Fig. 2 shows learning curves of the single PFLD, bagging and the RSM for the 80-dimensional data sets ($p+r$=80). Fig. 3 represents the generalization errors versus the number of redundant noise features r when the training sample size is fixed to 40 objects per class (80 training objects in total). Left plots in Fig. 2 and 3 show the case when the classification ability is concentrated in p features. Right plots represent the case when the discrimination power is spread over all $p+r$ features. Figures show that both combining techniques are useful in highly redundant feature spaces. The RSM performs relatively better when the discrimination power is spread over many features than when it is condensed in few features (this coincides with results obtained by Ho for decision trees [8]). When all data features are informative, the increasing redundancy in the data feature set does not affect the performance of the RSM. This can be explained as follows. In order to construct good classifiers in random subspaces, it is important that each subspace would contain as much as possible useful information. This could be achieved only when information is "uniformly" spread over all features (it is especially important when small random subspaces are used in the RSM). If useful information is condensed in few features, many random subspaces will be "empty" of useful information, the classifiers constructed in them will be bad and may worsen the combined decision.

In contrast to the RSM, the performance of bagging is affected neither by the redundancy representation in the data feature set, nor by the increasing feature redundancy (it is affected by the data dimensionality referred to the training sample size). It happens because all features are kept in bagging when training objects are sampled. Rotation does not change the informativity of the bootstrap replicate of the

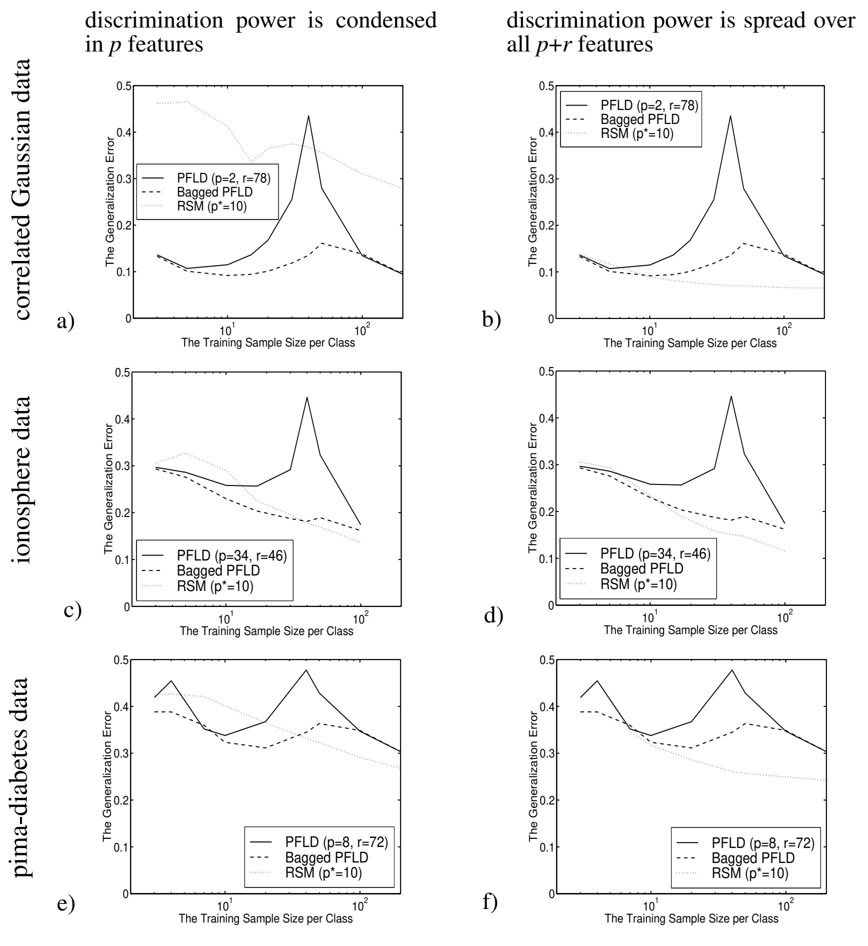

Fig. 2. Learning curves of the single PFLD, bagging and the RSM (p^*=10) for the 80-dimensional data sets ($p+r$=80). Left plots show the case when discrimination power is condensed in p features (data have r completely redundant noise features). Right plots show the case when discrimination power is spread over all features (after adding r redundant features to p-dimensional original data set, the data are rotated in all $p+r$=80 directions).

training set. Thus, bagging may perform better than the RSM for the highly redundant feature spaces when the discrimination power is condensed in few features and the training sample size is small. However, the RSM outperforms bagging when the discrimination power is distributed over all data features.

As well, besides the redundancy in the data feature set, other factors (e.g., the class overlap, the data distribution etc.) affect the performance of the PFLD, bagging and the RSM. Usually many factors act simultaneously, assisting and counteracting each other. Sometimes, some factors may have a stronger influence than the redundancy in the data feature set. For instance, the sonar data set represents time signals where the order of features jointly with their values are very important. When rotating this data set enriched by redundant noise features, some important information is lost. By this reason, the RSM performs worse on the rotated sonar data set than when no rotation is

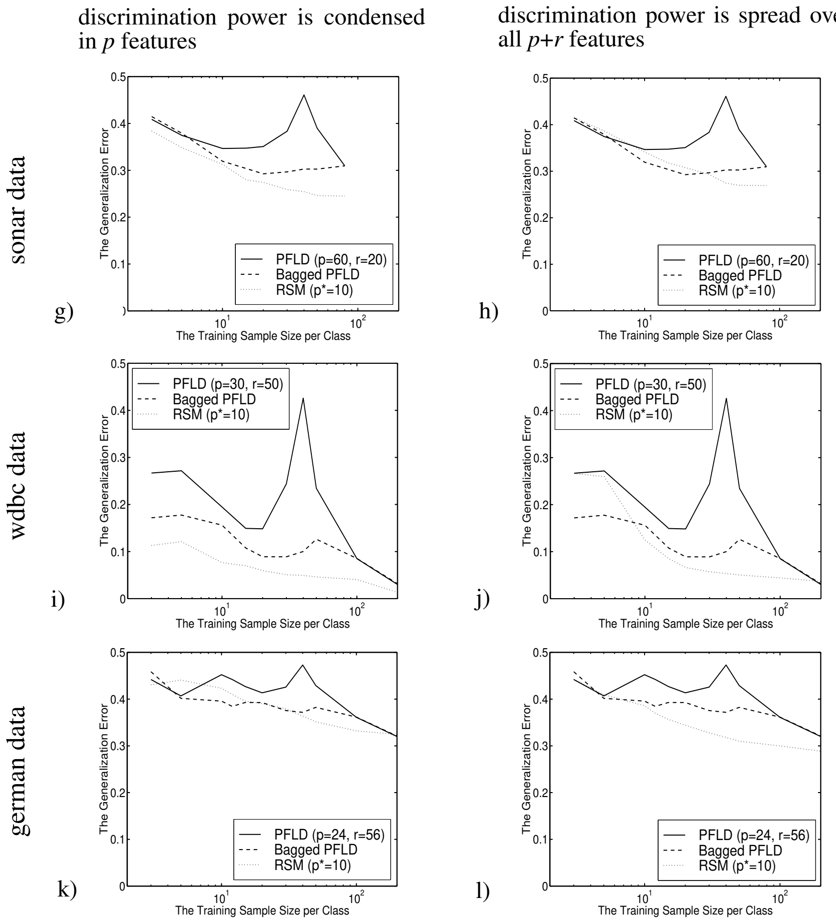

Fig. 2. Learning curves of the single PFLD, bagging and the RSM (p^*=10) for the 80-dimensional data sets (p+r=80). Left plots show the case when discrimination power is condensed in p features (data have r completely redundant noise features). Right plots show the case when discrimination power is spread over all features (after adding r redundant features to p-dimensional original data set, the data are rotated in all p+r=80 directions).

performed (see Fig. 2g,h and Fig. 3g,h). Another example, where rotation does not improve the performance of the RSM, is the modified wdbc data set (see Fig. 2i,j and Fig. 3i,j). All features of the original data set are already strongly correlated and it has the largest Mahalanobis distance (d=14.652) among all considered data sets. It seems that rotation somewhat "worsens" the distribution of the useful information over the features. So the RSM performs worse on the rotated data set than when no rotation is performed.

5 Conclusions

Besides the training sample size related to the data dimensionality and the choice of the base classifier, *the efficiency of the combining techniques may depend on the level*

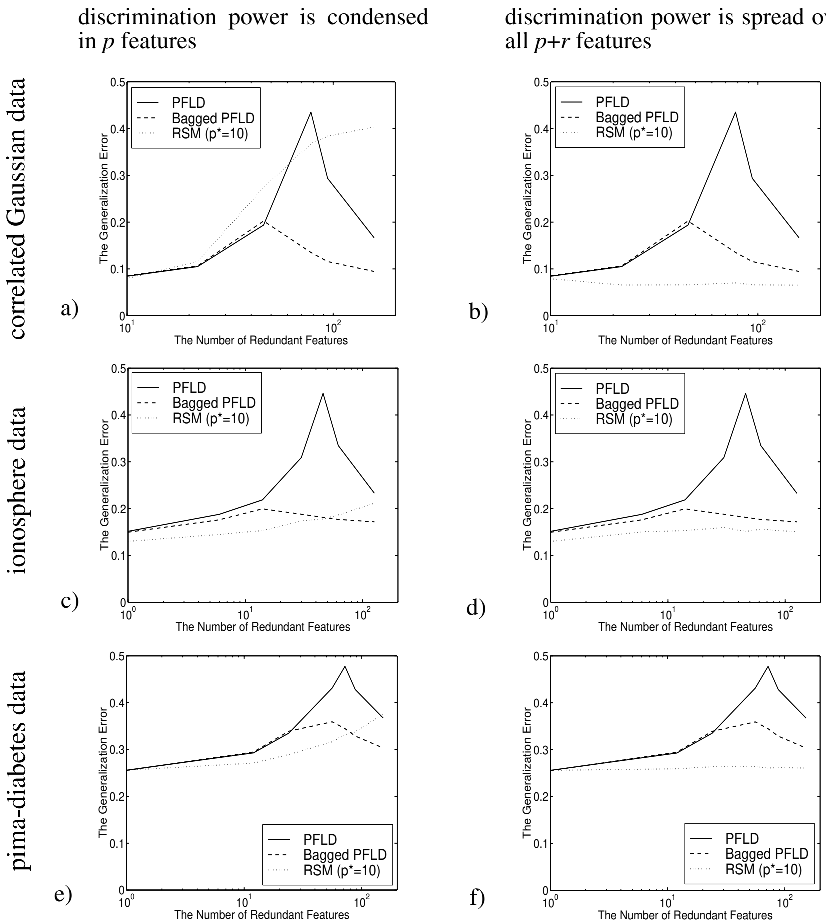

Fig. 3. The generalization errors of the single and the combined PFLD's versus the number of redundant noise features r added to the original p-dimensional data sets. Left plots show the case when a discrimination power is condensed in p features (data have r completely redundant noise features). Right plots represent the case when a discrimination power is spread over $p+r$ features (after adding r redundant features to p-dimensional data set, the data are rotated in all $p+r$ directions). 80 training objects are used to train the single and the combined classifiers.

of redundancy in the data feature set and on the way this redundancy is presented.

When applied to the Pseudo Fisher linear classifier, *the RSM performs relatively better when the classification ability* (discrimination power and also the redundancy) *is spread over many features* (i.e., for the data sets having many informative features) *than when the classification ability is condensed in few features* (i.e., for the data sets with many completely redundant noise features). *When the discrimination power is spread over all features, the RSM is resistant to the increasing redundancy in the data feature set*.

Unlike the RSM, *the performance of bagging*, when applied to the PFLD, *depends neither on the redundancy representation nor on the level of redundancy in the data*

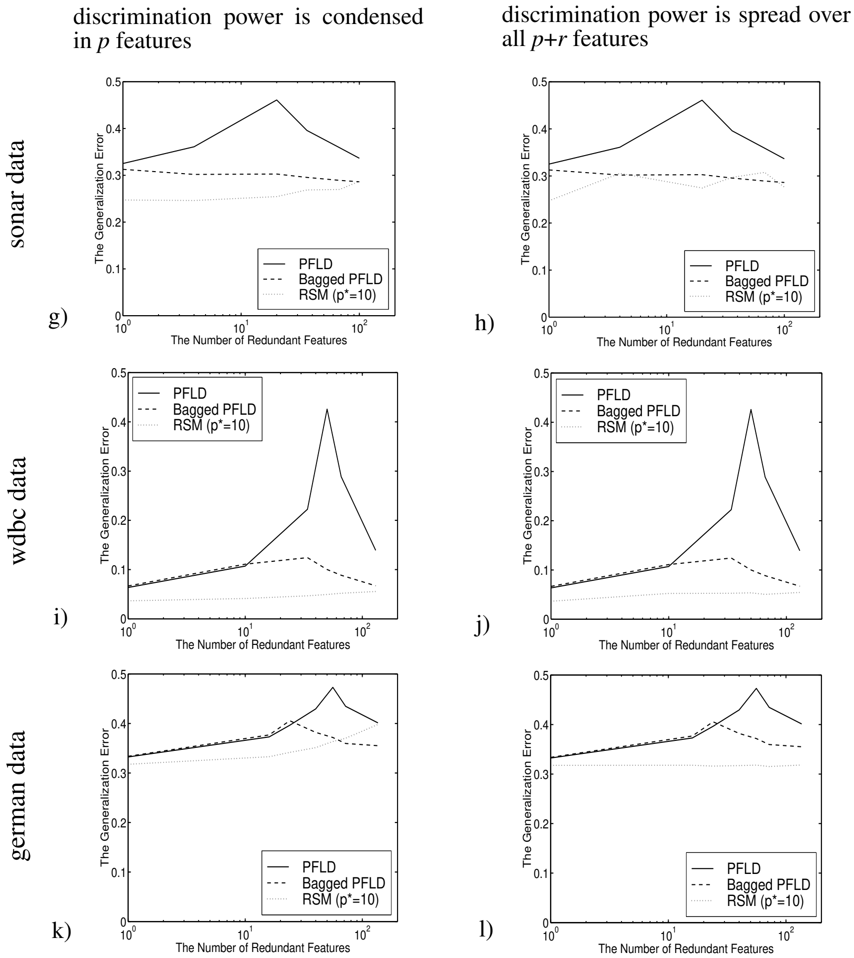

Fig. 3. The generalization errors of the single and the combined PFLD's versus the number of redundant noise features r added to the original p-dimensional data sets. Left plots show the case when a discrimination power is condensed in p features (data have r completely redundant noise features). Right plots represent the case when a discrimination power is spread over p+r features (after adding r redundant features to p-dimensional data set, the data are rotated in all p+r directions). 80 training objects are used to train the single and the combined classifiers.

feature set (it depends on the data dimensionality related to the training sample size). Therefore, *bagging may perform better than the RSM for the highly redundant feature spaces where the discrimination power is condensed in few features.* However, *when the discrimination power is spread over all features, the RSM outperforms bagging.*

The success of the combining techniques depends on many different factors that act simultaneously and may assist and counteract each other. However, notwithstanding the difficulty to study the influence of each factor independently of other ones, it is very important to understand what factors affect the performance of the combining techniques and in what way. Obviously, more study should be done in this direction.

Acknowledgment

This work is supported by the Foundation for Applied Sciences (STW) and the Dutch Organization for Scientific Research (NWO).

References

1. Jain, A.K., Chandrasekaran, B.: Dimensionality and Sample Size Considerations in Pattern Recognition Practice. In: Krishnaiah, P.R., Kanal, L.N. (eds.): Handbook of Statistics, Vol. 2. North-Holland, Amsterdam (1987) 835-855
2. Friedman, J.H.: Regularized Discriminant Analysis. JASA **84** (1989) 165-175
3. An, G.: The Effects of Adding Noise During Backpropagation Training on a Generalization Performance. Neural Computation **8** (1996) 643-674
4. Breiman, L.: Bagging predictors. Machine Learning Journal **24**(2) (1996) 123-140
5. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: Machine Learning: Proceedings of the Thirteenth International Conference (1996) 148-156
6. Ho, T.K.: The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(8) (1998) 832-844
7. Skurichina, M., Duin, R.P.W.: Bagging, Boosting and the Random Subspace Method for Linear Classifiers. Submitted to FMC special issue of Pattern Analysis and Applications (2001)
8. Ho, T.K.: Complexity of Classification Problems and Comparative Advantages of Combined Classifiers. In: Kittler, J., Roli, F. (eds.): Multiple Classifier Systems (Proceedings of the First International Workshop MCS 2000, Cagliari, Italy, June 2000). Lecture Notes in Computer Science, Vol. 1857, Springer-Verlag, Berlin (2000) 97-106
9. Fukunaga, K.: Introduction to Statistical Pattern Recognition. Academic Press (1990) 400-407
10. Skurichina, M., Duin, R.P.W.: The Role of Combining Rules in Bagging and Boosting. In Ferri, F.J., Inesta, J.M., Amin, A., Pudil, P. (eds): Advances in pattern recognition (Proceedings of the Joint Int. Workshops SSPR'2000 and SPR'2000, Alicante, Spain, Aug.-Sept. 2000). Lecture Notes in Computer Science, Vol. 1876, Springer-Verlag, Berlin (2000) 631-640
11. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman and Hall, New York (1993)
12. Skurichina, M., Duin, R.P.W.: Bagging for Linear Classifiers. Pattern Recognition **31**(7) (1998) 909-930
13. Skurichina, M., Duin, R.P.W.: Boosting in Linear Discriminant Analysis. In: Kittler, J., Roli, F. (eds.): Multiple Classifier Systems (Proceedings of the First International Workshop MCS 2000, Cagliari, Italy, June 2000). Lecture Notes in Computer Science, Vol. 1857, Springer-Verlag, Berlin (2000) 190-199
14. Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science (1998)
15. Golomb, S.W., Baumert, L.D.: The Search for Hadamard Matrices, American Mathematics Monthly **70** (1963) 12-17