Logistic Ensembles for Principal Direction and Random Spherical Linear Oracles

Leif E. Peterson, Matthew A. Coleman

Abstract—**Principal direction linear oracle (PDLO) and random spherical linear oracle (RSLO) ensemble classifiers for DNA microarray gene expression data are proposed. The oracle assigns different training(testing) samples to 2 sub-classifiers of the same type using hyperplane splits in order to increase the diversity of voting results since errors are not shared across sub-classifiers. Eleven classifiers were evaluated for performance as the base classifier including k nearest neighbor (kNN), naïve Bayes classifier (NBC), linear discriminant analysis (LDA), learning vector quantization (LVQ1), polytomous logistic regression (PLOG), artificial neural networks (ANN), constricted particle swarm optimization (CPSO), kernel regression (KREG), radial basis function networks (RBFN), gradient descent support vector machines (SVMGD), and least squares support vector machines (SVMLS). Logistic ensembles (PLOG) resulted in the best performance when used as a base classifier for PDLO and RSLO. The performance of logistic ensembles based on random axis selection and rotation followed by hyperplane splits in PDLO increased with increasing CV-fold and iteration number. However, logistic ensembles employed with random hyperplane splits used in RSLO resulted in degeneration of performance at the greatest levels of CV-fold and iteration number.**

I. Introduction

Promising results have been reported for ensemble classification methods for which moderate to substantial gains in performance have been realized [1]. Ensemble methods combine results of multiple classifiers to exploit complex decision boundaries that cannot be handled by a single classifier. The most common methods include majority voting [2], mixture of experts [3]–[6], bagging [7], boosting [8]–[11], and bootstrapping [12].

Random oracles for classification originated in the development of oblique decision trees in the form of axis-parallel splits used in CART-LC [13] and later OC1 [14]. Ensembles of random linear oracles were recently applied to 35 UCI data sets using a variety of methods including Adaboost, bagging, multiboost, random subspace, and random forests, and were shown to outperform other ensemble methods [15], [16]. The main utility of an oracle for classification is to randomly split and assign training and testing samples to two sub-classifiers which employ the same base classifier (decision tree, NBC, kNN, etc.) for training. Following training, each sub-classifier performs class prediction on the assigned test samples left out of training. The primary advantages for using a linear oracle is that the training and testing of samples partitioned into two disjoint partitions can be easier than jointly performing the same operations on the combined samples, and secondly, more diversity is introduced into training and testing because potential errors made by the sub-classifiers are not shared [17]. Diversity has been investigated for boosting, bagging, sum and product rule for majority voting, etc., [18], [19]; however, to date there exists no unifying theory that can predict the relationship between ensemble diversity and classification error [20]. Not surprisingly, most efforts to overproduce-and-select the best combination of classifiers for a specific classification problem have revealed that finding an optimum collection of classifiers is not as straightforward as one would assume.

In a recent report on ensemble classification of gene expression [21], we showed that a principal direction linear oracle (PDLO) with a logistic base classifier outperformed logistic classification without PDLO, and that PDLO improved the performance of logistic classifiers the most when compared with ten other base classifiers (kNN, NBC, LDA, LVQ1, ANN, CPSO, KREG, RBFN, SVMGD, and SVMLS). During PDLO, assignment of training and testing samples to the sub-classifier by the oracle is based on rotation of normalized (spherical) principal component (PC) scores about the eigenvectors representing the first, second, and third principal components extracted from the input samples and features. The eigenvectors for the first 3 principal components represent orthogonal X, Y, and Z axes in three-dimensional space which can be rotated using a combinations of fixed or random selections of axes and rotation angles. The rationale for using PDLO was that, in the case of the first principal component (i.e., greatest eigenvalue), the rotation of PC scores around the first eigenvector would ensure uniform selection of samples by the oracle, since the first eigenvector will disect the centroids of data clusters with the most separation. A spherical projection of the PC scores for the first 3 eigenvectors (X, Y, Z) was induced by standardizing (normalizing) with $\sqrt{\lambda_l}$, where λ_l is the eigenvalue for the lth component. Hyperplane splits of samples were achieved by selecting cases with positive(negative) PC scores after rotation of the specific axis about which scores were rotated.

This report compares performance of base classifiers for RSLO against performance of base classifiers used with PDLO, and in addition, compares performance results for logistic ensembles for PDLO and RSLO as a function of CV-folds and iterations (hyperplane splits). Empirical gene expression data are used for the analysis.

L.E. Peterson is with the Division of Biostatistics and Epidemiology, Dept. of Public Health, The Methodist Hospital, 6550 Fannin Street, SM-1299, Houston, Texas 77030, USA. E-mail: peterson.leif@ieee.org.

M.A. Coleman is with the Biology and Biotechnology Research Program, Lawrence Livermore National Laboratory, 7000 East Avenue, Livermore, California 94550, USA. E-mail: coleman16@llnl.gov.

TABLE I

DATA SETS USED FOR CLASSIFICATION ANALYSIS.

Cancer	Classes-Genes-Samples	Selected*
Brain [22]	2-7129-60 (21 censored, 39 failures)	16
Prostate [23]	2-12600-102 (52 tumor, 50 normal)	11
Breast [24]	2-3170-15 (8 BRCA1, 7 BRCA2)	6
Breast [25]	2-24481-78 (34 relapse, 44 non-relapse)	17
Colon [26]	2-2000-62 (40 negative, 22 positive)	5
Lung [27]	2-12533-32 (16 MPM, 16 ADCA)	29
Leukemia [28]	2-7129-38 (27 ALL, 11 AML)	9
Leukemia [29]	3-12582-57 (20 ALL, 17 MLL, 20 AML)	13
SRBCT [30]	4-2308-63 (23 EWS, 8 BL, 12 NB, 20 RMS)	20

* Genes selected using greedy PTA.

II. Methods

A. DNA Microarray Data Sets Used

Data used for classification analysis were available in C4.5 format from the Kent Ridge Biomedical Data Set Repository (http://sdmc.i2r.a-star.edu.sg/rp), see Table I. The 2-class adult brain cancer data were comprised of 60 arrays (21 censored, 39 failures) with expression for 7,129 genes [22]. The 2-class adult prostate cancer data set consisted of 102 training samples (52 tumor, and 50 normal) with 12,600 features. The original report for the prostate data supplement was published by Singh et al [23]. Two breast cancer data sets were used. The first had 2 classes and consisted of 15 arrays for 8 BRCA1 positive women and 7 BRCA2 positive women with expression profiles of 3,170 genes [24], and the second was also a 2-class set including 78 patient samples and 24,481 features (genes) comprised of 34 cases with distant metastases who relapsed ("relapse") within 5 years after initial diagnosis and 44 disease-free ("non-relapse") for more than 5 years after diagnosis [25]. Two-class expression data for adult colon cancer were based on the paper published by Alon et al [26]. The data set contains 62 samples based on expression of 2000 genes in 40 tumor biopsies ("negative") and 22 normal ("positive") biopsies from non-diseased colon biopsies from the same patients. An adult 2-class lung cancer set including 32 samples (16 malignant pleural mesothelioma (MPM) and 16 adenocarcinoma (ADCA)) of the lung with expression values for 12,533 genes [27] was also considered. Two leukemia data sets were evaluated: one 2-class data set with 38 arrays (27 ALL, 11 AML) containing expression for 7,129 genes [28], and the other consisting of 3 classes for 57 pediatric samples for lymphoblastic and myelogenous leukemia (20 ALL, 17 MLL and 20 AML) with expression values for 12,582 genes [29]. The Khan et al [30] data set on pediatric small round blue-cell tumors (SRBCT) had expression profiles for 2,308 genes and 63 arrays comprising 4 classes (23 arrays for EWS-Ewing Sarcoma, 8 arrays for BL-Burkitt lymphoma, 12 arrays for NB-neuroblastoma, and 20 arrays for RMS-rhabdomyosarcoma).

B. Gene Filtering and Selection

For each data set, input genes were ranked by the F-ratio test statistic, and the top 150 were then used for gene selection. Gene selection was based on a stepwise greedy plus-take-away (PTA) method using a plus 1 take away 1 heuristic [31]. Gene-specific expression on each array was standardized using the mean and standard deviation over the 150 genes identified by filtering. Forward stepping was carried out to add(delete) the most(least) important genes for class separability based on squared Mahalanobis distance and the F-to-enter and F-remove statistics. Genes were entered into the model if their standardized expression resulted in the greatest Mahalanobis distance between the two closest classes and their F-to-enter statistic exceeded the F-to-enter criterion. At any step, a gene was removed if its F-to-enter statistic (F=3.84) was less than the F-to-remove criterion (F=2.71). Table I lists the number of genes selected using greedy PTA.

C. Principal Direction Linear Oracle (PDLO)

Let $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ be the set of feature values for sample $\mathbf{x}_i$, and $\mathbf{z}_i = (z_{i1}, z_{i2}, \dots, z_{ip})$ be the set of standardized feature values for sample $\mathbf{x}_i$. Let $\mathbf{R}$ be the $p \times p$ ("gene by gene") correlation matrix based on n training samples. The concept of principal directions relies on the eigenvectors derived from principal component analysis (PCA). Let $\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_m$ represent the eigenvectors associated with the m greatest eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$ extracted from $\mathbf{R}$. Principal components derived from PCA are orthogonal (uncorrelated) with one another. There are 3 vectors associated with each lth extracted principal component ($l = 1, 2, \dots, m$): a p-vector eigenvector $\mathbf{e}_l$, a p-vector of principal component score coefficients $\boldsymbol{\beta}_l$ determined with the relationship $\beta_{jl} = e_{jl}/\sqrt{\lambda_l}$, and an n-vector of PC scores $\mathbf{y}_l$, where the score for the ith ($i = 1, 2, \dots, n$) sample given as

$$y_{il} = \beta_{1l} z_{i1} + \beta_{2l} z_{i2} + \cdots + \beta_{pl} z_{ip}. \tag{1}$$

The vector $\mathbf{y}_l$ is distributed $N(0, 1)$ and serves as a new feature representing each sample in score space. PC scores for the first 3 principal components are then used for representing each sample in spherical 3D space by defining the point $\boldsymbol{q}_i = (X_i, Y_i, Z_i)$, where $X_i = y_{i1}$, $Y_i = y_{i2}$, and $Z_i = y_{i3}$. Depending on the axis selected, a hyperplane split of all points rotated about an eigenvector is made. For example, if points are rotated about the X-axis, then a hyperplane $h(\boldsymbol{Y})$ based on $Y_i = 0$ can be used for splitting the samples. Points having zero or positive values of Y_i will lie on or above $h(\boldsymbol{Y})$ and are assigned to data set $\mathcal{D}_1$, whereas samples with negative Y_i lie below $h(\boldsymbol{Y})$ and are assigned to $\mathcal{D}_2$. The first miniensemble is used for training and testing with $\mathcal{D}_1$ and the second miniensemble used for training and testing with $\mathcal{D}_1$. The predicted class membership of test samples in $\mathcal{D}_1$ and $\mathcal{D}_2$ are then used during construction of the confusion matrix used in performance evaluation.

PDLO performance was evaluated for two methods of rotation, one in which an increasing angle of rotation was used where $\theta = \text{iteration}\frac{2\pi}{\#\text{iterations}}$ and another involving randomly selected rotation angles with $\theta = U(0, 1)2\pi$. Algorithm 1 lists the computational steps for employing the principal direction linear oracle to invoke a hyperplane to split of samples into a set $\mathcal{D}_1$ above a hyperplane and a set $\mathcal{D}_2$ below.

D. Random Spherical Linear Oracle (RSLO)

The random spherical linear oracle (RSLO) used the same spherical 3D points developed during PDLO; however, axis-

Algorithm 1: Principal Direction Hyperplane Splits

```
Data: $X$-, $Y$-, or $Z$-axis in spherical 3D score space for
      the 3 greatest eigenvalues for a set of $p$ training
      (testing) genes selected with greedy PTA.
Result: A set of samples, $\mathcal{D}_1$, on or above a hyperplane
        and a set $\mathcal{D}_2$ below the hyperplane
Specify rotation axis $V$ as either the $X$-, $Y$-, or $Z$-axis
Select axis $W$ which is orthogonal to $V$-axis
Select whether rotation angles are fixed intervals or
random
foreach iteration $j$ do
    If rotation angles are fixed intervals: $\theta = j \frac{2\pi}{\#\text{iterations}}$
    If rotation angle is random: $\theta = U(0,1)2\pi$
    for sample $i \leftarrow 1$ to $n$ do
        Rotate point $\boldsymbol{q}_i$ around axis $V$ using angle $\theta$
        if $W_i \geq 0$ then
            Sample is on or above hyperplane
            $h(\boldsymbol{W}) \geq 0 \rightsquigarrow \mathcal{D}_1$
        endif
        if $W_i < 0$ then
            Sample is below hyperplane $h(\boldsymbol{W}) < 0 \rightsquigarrow \mathcal{D}_2$
        endif
    endfor
endfch
```

independent random hyperplane splits were employed. Using the points $\boldsymbol{q}_i$ in 3D score space, random hyperplane splits were performed by calculating the dot product $\boldsymbol{q}\boldsymbol{\mu}$ for each point, where $\boldsymbol{\mu}$ is a unit vector with direction cosines μ_X, μ_Y, and μ_Z for a randomly selected point $\boldsymbol{q}_{U(0,1)(n-1)+1}$. Points for which $\boldsymbol{q}\boldsymbol{\mu} \geq 0$ lie on or above the hyperplane $h(\boldsymbol{q}\boldsymbol{\mu} = 0)$ and are assigned to data set $\mathcal{D}_1$, whereas points with dot product $\boldsymbol{q}\boldsymbol{\mu} < 0$ lie below the hyperplane and are assigned to $\mathcal{D}_2$.

E. Identification of Best Base Classifier

The first step during classification analysis was to identify a that yielded the best performance levels. Eleven classifiers were employed for training and testing [32], [33]. These included k nearest neighbor (kNN), naïve Bayes classifier (NBC), linear discriminant analysis (LDA), learning vector quantization (LVQ1), polytomous logistic regression (PLOG), artificial neural networks (ANN), constricted particle swarm optimization (CPSO), kernel regression (KREG), radial basis function networks (RBFN), gradient descent support vector machines (SVMGD), and least squares support vector machines (SVMLS). KREG employed kernel tricks in a least squares fashion to determine coefficients which reliably predict class membership when multiplied against kernels for test samples. All 2-class and 3-class problems were solved using all possible 2-class problems. First, k-means cluster analysis was performed on all of the training samples to determine centers. Coefficients for kernel regression were determined using the least squares model

$$\boldsymbol{\alpha} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}, \tag{2}$$

where $\mathbf{H}$ is a sample × gene matrix with a linear kernel in element $h_{ij} = K(\mathbf{x}_i, \mathbf{c}_j) = \mathbf{x}_i^T\mathbf{c}_j$, where $\mathbf{c}_j$ is a center vector, i.e., mean vector, from k-means cluster analysis, and $\mathbf{y}$ is sample vector with y_i set to +1 for training samples in the first class and -1 for samples in the second class being compared in the 2-class problem. A positive value of y_i denotes membership in the first class and a negative value reflects membership in the second class. The RBFN employed the same matrix algebra as kernel regression, but was based on the RBF kernel $K(\mathbf{x}_i, \mathbf{c}_j) = \exp(-||\mathbf{x}_i - \mathbf{c}_j||)$. Note that this is not a Gaussian RBF kernel which uses $\exp(-||\mathbf{x}_i - \mathbf{c}_j||/\sigma)$ as the kernel. For SVMs, we used an L1 soft norm gradient descent-based [34] and L2 soft norm least squares approach to SVM [35]. A weighted exponentiated RBF kernel was employed to map samples in the original space into the dot-product space, given as $K(\mathbf{x}, \mathbf{x}^T) = \exp(-\frac{\gamma}{m}||\mathbf{x} - \mathbf{x}^T||)$, where m=#features. Such kernels are likely to yield the greatest class prediction accuracy providing that a suitable choice of γ is used. To determine an optimum value of γ for use with RBF kernels, a grid search was done using incremental values of γ from 2^{-15}, 2^{-13},..., 2^3 in order to evaluate accuracy for all training samples. We also used a grid search in the range of 10^{-2}, 10^{-1},..., 10^4 for the SVM margin parameter C. The optimal choice of C was based on the grid search for which classification accuracy is the greatest, resulting in the optimal value for the separating hyperplane and minimum norm $||\xi||$ of the slack variable vector. SVM tuning was performed by taking the median of parameters during grid search iterations when the test sample misclassification rate was zero. Base classifier performance for PDLO and RSLO was evaluated using ten 10-fold CV [36].

F. Effect of CV and Iterations on PDLO and RSLO

PLOG yielded the greatest performance when implemented with PDLO and RSLO, and therefore was evaluated using CV folds of 2, 5, and 10, and 10 to 100 iterations.

III. Results

Figure 1 and Figure 2 show boxplots of classifier accuracy for different base classifiers implemented with PDLO and RSLO, respectively. For PDLO, LDA and PLOG yielded the greatest performance, and for RSLO PLOG had the greatest performance. Figure 3 and Figure 4 illustrate for 2-fold CV (CV2) scatter plots with the number of training (testing) samples assigned to subset $\mathcal{D}_1$ on or above hyperplanes and samples in $\mathcal{D}_2$ below hyperplanes for PDLO and RSLO, respectively. It is readily apparent that different slopes (training to testing sample ratios) occurred for samples assigned to $\mathcal{D}_1$ and $\mathcal{D}_2$ during random hyperplane splits in RSLO. For PDLO, the slopes for training/testing sample ratios for samples assigned to $\mathcal{D}_1$ and $\mathcal{D}_2$ were similar. Finally, Figure 5 and Figure 6 reveal that performance from axis-dependent rotations followed by hyperplane splits in PDLO increased with the number of iterations and number of CV-folds. On the other hand, for random hyperplane splits in RSLO, this trend degenerated.

IV. Discussion and Conclusion

There is currently a limited body of information on oracle ensemble classification for gene expression data. To date,

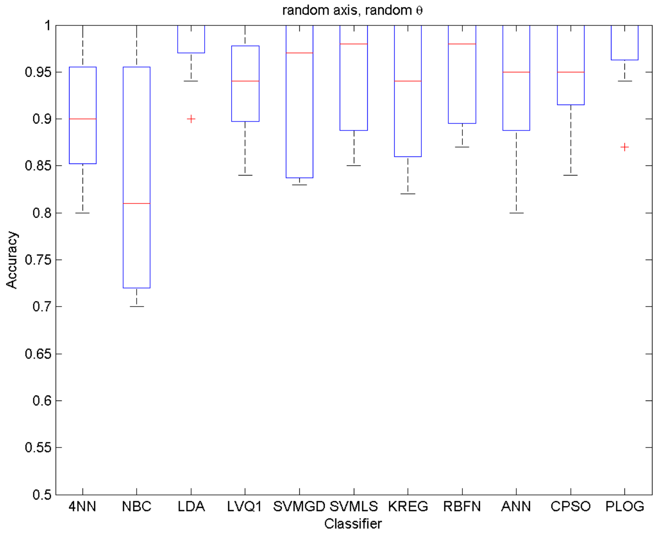

Fig. 1. Principal direction linear oracle (PLDO) classifier performance (accuracy) for various base classifiers.

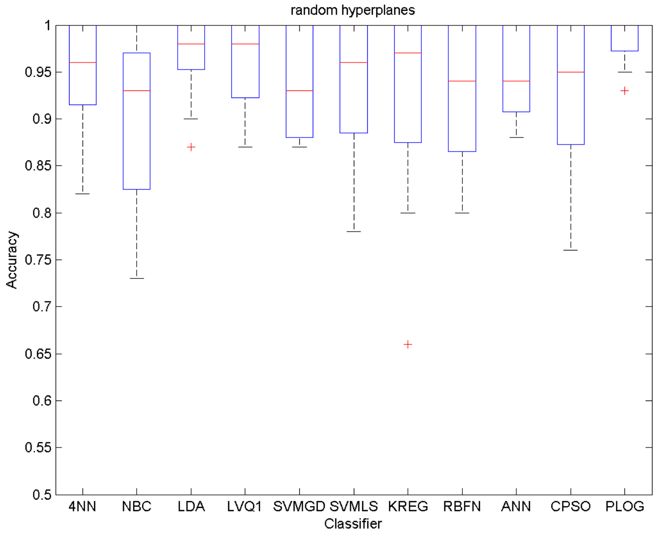

Fig. 2. Random spherical linear oracle (RSLO) classifier performance (accuracy) for various base classifiers.

the majority of work has been performed by Kuncheva and Rodriquez using random linear oracles based on J48 decision trees applied to UCI data sets [15], and spherical oracles based on Naïve Bayes base classifiers [16]. For the most part, their choice of feature transformation involved normalizing (rescaling) all feature values in the range {0,1} and performing random hyperplane splits by selecting 2 points randomly and defining the hyperplane from the perpendicular vector at the midpoint between the 2 points. Superior results were obtained for the random linear oracle when compared with the routine uses of various bagging and boosting forms of decision tree methods.

In the present study, our focus was to evaluate the effect of axis-dependent rotation and hyperplane splits in PDLO

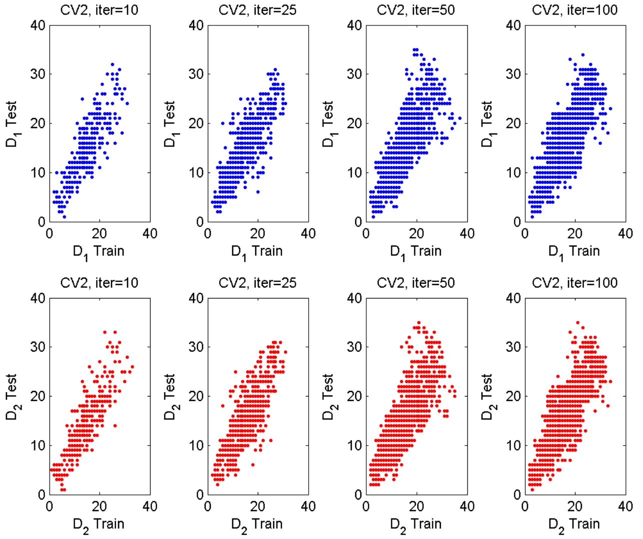

Fig. 3. Number of training (testing) samples assigned to subset $\mathcal{D}_1$ on or above hyperplanes vs. samples below hyperplanes in $\mathcal{D}_2$ for PDLO based on random axis selection and random rotation angle selection prior to hyperplane splits. Test samples left out of training during CV.

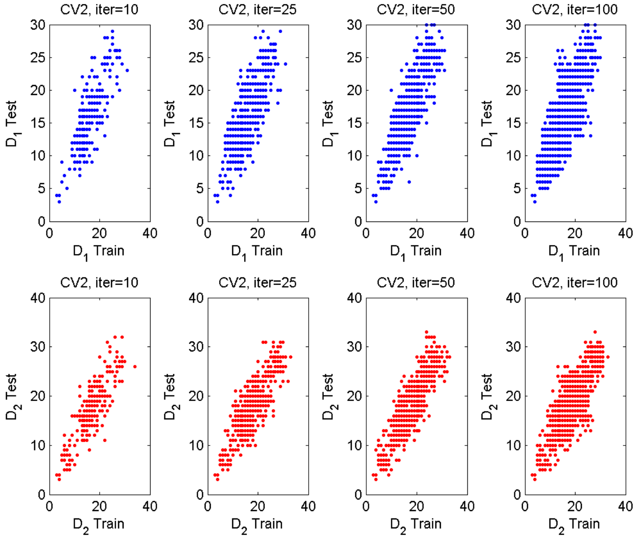

Fig. 4. Number of training (testing) samples in $\mathcal{D}_1$ on or above hyperplanes vs. samples in $\mathcal{D}_2$ below hyperplanes for RSLO based on random hyperplane splits. Test samples left out of training during CV.

and random hyperplane splits in RSLO on performance for 11 base classifiers. The use of orthogonal eigenvectors for the 3 greatest eigenvalues of the gene-by-gene correlation matrix of expression, normalized with $\sqrt{\lambda_l}$, provided a spherical 3D set of points in score space about which rotations were carried out. Our results show that logistic ensembles resulted in the greatest classification performance for PDLO and RSLO. Moreover, the performance of logistic ensembles based on axis-dependent rotations followed by hyperplane splits in PDLO increased with increasing CV-fold and iteration number. However, logistic ensembles employed with random hyperplane splits used in RSLO resulted in degeneration of

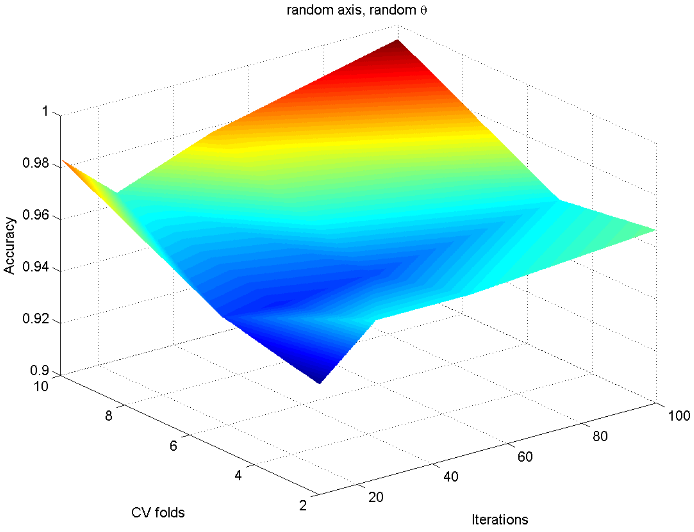

Fig. 5. PDLO accuracy as a function of CV folds and iterations. Each iteration based on axis-dependent rotation of points followed by hyperplane splits. PLOG used as base classifier.

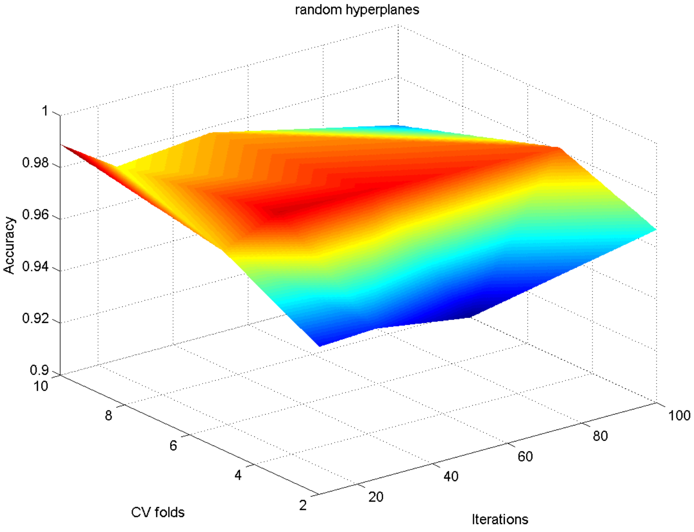

Fig. 6. RSLO accuracy as a function of CV folds and iterations. Each iteration based on random hyperplane split. PLOG used as base classifier.

performance at the greatest levels of CV-fold and iteration number.

In conclusion, PLOG resulted in the best performance when used as a base classifier for PDLO and RSLO. For 10-fold CV and 100 iterations, the axis-dependent rotations and hyperplane splits used in PDLO resulted in greater performance levels when compared with random hyperplane splits used in RSLO.

References

[1] J. Kittler, M. Hatef, R.P.W. Duin, J. Matas. On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions, 20(3), 226-239. 1998

[2] van Erp, M., Vuupijl, L., and Shomaker, L. An overview and comparison of voting methods for pattern recognition. 1-6. 2002. Hoboken(NJ), IEEE. Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (WFHR02).

[3] M.I. Jordan, R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. Neural Computation, 6, 181-214, 1994.

[4] S. Gutta, J.R.J. Huang, P. Jonathon, H. Wechsler. Mixture of experts for classification of gender, ethnic origin, and pose of human faces. IEEE Transactions on Neural Networks. 11(4), 948-960, 2000.

[5] K. Chen, L. Xu, H. Chi. Improved learning algorithms for mixture of experts in multiclass classification. Neural Networks. 12, 1229-1252 (1999).

[6] A. Rao, D. Miller, K. Rose, and A. Gersho. Mixture of experts regression modeling by deterministic annealing. IEEE Transactions on Signal Processing. 45(11), 2811-2820, 1997.

[7] L. Breiman. Bagging predictors. Machine Learning. 24(2), 123-140, 1996.

[8] Y. Freund, R.E. Schapire. Experiments with a New Boosting Algorithm, in Proceedings of the Thirteenth International Conference on Machine Learning, 148-156. San Francisco, Morgan Kaufmann, 1996.

[9] Y. Freund, R.E. Schapire. A short introduction to boosting. Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, 1999.

[10] Qu, Y, Adam, BH, Yasuo, Y, Ward, MD, Cazres, LH, Schellhammer, PF, Feng, Z., Semmes, O.J., Wright, GL. Boosted decision tree analysis of surface-enhanced laser desorption/ionization mass spectral serum profiles discriminates prostate cancer from noncancer patients. Clinical Chemistry 2002. 48(10): 1835-1843.

[11] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos. Boosting linear discriminant analysis for face recognition. ICIP (1), 657-660, 2003.

[12] Efron, B. Estimating the error rate of a prediction rule: Improvement on cross-validation. J. American Stat. Assoc. 1983. 78:316-331.

[13] L. Breiman, J. Friedman, R. Olshen, C. Stone. Classification and Regression Trees. Boca Raton(FL), Chapman & Hall/CRC, 1984.

[14] Murthy, S.K., S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. Journal of Artificial Intelligence Research. 2, 1-33, 1994.

[15] L.I. Kuncheva, J.J. Rodriguez. Classifier Ensembles with a Random Linear Oracle. IEEE Transactions on Knowledge and Data Engineering. 19(4) 500-508, 2007.

[16] J.J. Rodriguez, Kuncheva. Naive Bayes ensembles with a random oracle. Lecture Notes in Computer Science. 4472: 450-458, 2007.

[17] A.J.C. Sharkey, N.E.Sharkey, Combining diverse neural net. The Knowledge Engineering Review. 12(3)231-247, 1997.

[18] Kuncheva LI, Whitaker CJ. Measures of diversity in classifier ensembles. Machine Learning 2003;51:181-207.

[19] Kuncheva LI. That Elusive Diversity in Classifier Ensembles. Lecture Notes in Computer Science 2003;2652:1126-38.

[20] Brown, G., Wyatt, J., Harris, R., Yao, X. Diversity creation methods: a survey and categorisation. Information Fusion. 6(1), 5-20, 2005.

[21] L.E. Peterson, Coleman, M.A. Principal direction linear oracle for gene expression ensemble classification. Workshop on Computational Intelligence Approaches for the Analysis of Bioinformatics Data (CIBIO07); 2007 Int. Joint Conference on Neural Networks (IJCNN07), 2007.

[22] S.L. Pomeroy, P. Tamayo, M. Gaasenbeek, L.M. Sturla, M. Angelo, M.E. McLaughlin, J-Y.H. Kim, L.C. Goumnerovak, P. M. Blackk, C. Lau, J.C. Allen, D. Zagzagl, J.M. Olson, T. Curran, C. Wetmore, J.A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califanokk, G. Stolovitzkykk, D.N. Louis, J.P. Mesirov, E.S. Lander, T.R. Golub, Prediction of central nervous

system embryonal tumour outcome based on gene expression, Nature. 415(6870) (2002) 436-442.

[23] D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, J. Manola, C. Ladd, P. Tamayo, A.A. Renshaw , A.V. D'Amico, J.P. Richie, E.S. Lander, M. Loda, P.W. Kantoff, T.R. Golub, W.R¿ Sellers, Gene expression correlates of clinical prostate cancer behavior, Cancer Cell. 1(2) (2002) 203-209.

[24] I. Hedenfalk, D. Duggan, Y. Chen et al, Gene-expression profiles in hereditary breast cancer, N. Engl. J. Med. 344 (2001) 539-548.

[25] L.J. van 't Veer, H. Dai, M.J. van de Vijver, Y.D. He, A.A. Hart, M. Mao, H.L. Peterse, K. van der Kooy, M.J. Marton, A.T. Witteveen, G.J. Schreiber, R.M. Kerkhoven, C. Roberts, P.S. Linsley, R. Bernards, S.H. Friend, Gene expression profiling predicts clinical outcome of breast cancer, Nature. 415 (2002) 530-536.

[26] U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack, and A.J. Levine, Broad patterns of gene expression revealed by clustering of tumor and normal colon tissues probed by oligonucleotide arrays, Proc. Natl. Acad. Sci. USA. 96(12) (1999) 6745-6750.

[27] G.J. Gordon, R.V. Jensen, L.L. Hsiao, S.R. Gullans, J.E. Blumenstock, S. Ramaswamy, W.G. Richards, D.J. Sugarbaker, R. Bueno, Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma, Cancer Res. 62(17) (2002) 4963-5967.

[28] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander, Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression, Science. 286 (1999) 531-537.

[29] S.A. Armstrong, J.E. Staunton, L.B. Silverman, R. Pieters, M.L. den Boer, M.D. Minden, S.E. Sallan, E.S. Lander, T.R. Golub, S.J. Korsmeyer, MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia, Nature Genetics. 30(1) (2001) 41-47.

[30] J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu CR, C. Peterson, R.S., Meltzer, Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nature Med., 7 (2001) 673-679.

[31] Somol P., Pudil P., Nonovicova J., Paclik J. Adaptive floating search methods in feature selection. Pattern Recognition Letters 1999;20:1157-63.

[32] Peterson LE, Coleman MA. Machine learning-based receiver operating characteristic (ROC) curves for crisp and fuzzy classification of DNA microarrays in cancer research. Int. J. Approx. Reasoning, (in press).

[33] Peterson, L.E., Hoogeveen, R.C., Pownall, H.J., Morrisett, J.D. Classification Analysis of Surface-enhanced Laser Desorption/Ionization Mass Spectral Serum Profiles for Prostate Cancer. Proceedings of the 2006 IEEE World Congress on Computational Intelligence (WCCI 2006).

[34] Christianini, N. and Shawe-Taylor, J. 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge Univ. Press, Cambridge.

[35] Abe, S. Support Vector Machines for Pattern Classification. Advances in Pattern Recognition Series. Springer, Berlin, 2005.

[36] Kohavi R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. International Joint Conference on Artificial Intelligence (IJCAI) 1995;1137-45.