Classification of high-dimensional data with ensemble of logistic regression models

Noha Lim[1], Hongshik Ahn[2*], Hojin Moon[3]

and James J. Chen[4]

[1]GlaxoSmithKline, King of Prussia, PA 19406

[2]Department of Applied Math and Statistics
Stony Brook University, Stony Brook, NY 11794-3600
hahn@ams.sunysb.edu, Phone: (631) 632-8372, Fax: (631) 632-8490

[3]Department of Mathematics and Statistics
California State University, Long Beach, CA 90840-1001

[4]Division of Personalized Nutrition and Medicine, Biometry Branch
National Center for Toxicological Research, Jefferson, AR 72079

*Corresponding author: Hongshik Ahn

The views presented in this article are those of the authors and do not necessarily represent those of the U.S. Food and Drug Administration.

Abstract

A classification method is developed based on ensembles of logistic regression models, with each model fitted from a different set of predictors determined by a random partition of the feature space. The proposed method enables class prediction by an ensemble of logistic regression models for a high-dimensional data set, which is impossible by a single logistic regression model due to the restriction that the sample size needs to be larger than the number of predictors. The proposed classification method is applied to gene expression data on pediatric AML patients to predict patient's risk for treatment failure or relapse at the time of diagnosis. Hence, specific prognostic biomarkers can be used to predict outcomes in pediatric AML and formulate individual risk-adjusted treatment. Our study shows that the proposed method is comparable to other widely used models in generalized accuracy and is significantly improved in balance between sensitivity and specificity. The proposed ensemble algorithm enables the standard classification model to be used for classification of high-dimensional data.

Key Words: Aggregation; Class prediction; Cross validation; Decision threshold; Majority voting, Random partition.

1 Introduction

We introduce a classification procedure which is based on ensembles of logistic regression models. In the proposed method, each base classifier is constructed from a different set of predictors determined by a random partition of the entire set of predictors. This method is used for a classification of high-dimensional data. The proposed approach is based on a widely used standard regression model for binary responses, and it is less computer-intensive than tree-based ensemble methods.

Clinical practice strives to provide the best treatment. Ideally, patient treatment should be based on individual's disease characteristics and risk factors. It is often hypothesized that an individual's levels of gene transcripts reflect disease characteris-

tics or risk factors, and microarray gene expression profiles area is a promising source of genomic biomarkers, which could guide clinical practice.

Recent advancements in biotechnology have accelerated the search for molecular biomarkers useful in the diagnosis and treatment of disease. Molecular biomarkers of disease risk and status are critical to an accurate treatment by identifying patients most likely to benefit from particular drugs or patients most likely to experience adverse reactions. Because medicine is always practiced on individuals rather than populations, the goal is to change the assignment of therapies from population-based approach to an individualized approach.

Classification algorithms are an essential tool for identifying potential genomic biomarkers in high-dimensional data. In this article we introduce Logistic Regression Ensembles (LORENS) for classification of high-dimensional data. A full logistic regression model is used in each subset of the partition in LORENS. The predicted values from all the subsets are averaged for the ensemble classification. The conventional logistic regression model requires a variable selection if the number of predictors exceeds the sample size. However, in applications where large dimensionality is common, the standard model selection procedures become computationally not feasible given the huge dimension of the feature space. In LORENS, we can avoid variable selection by randomly partitioning the feature space into mutually exclusive subspaces. LORENS is a natural improvement over the logistic regression which is straightforward and whose algorithm is easy to implement. By using LORENS, we can improve the prediction accuracy, and resolve the issue of having more predictors than the sample size in logistic regression. LORENS employs the CERP (Classification by Ensembles from Random Partitions) methodology (Ahn et al., 2007) in constructing an ensemble of logistic regression models. CERP produces an ensemble of tree classifiers, each constructed from a different set of predictors. CERP combines the results of multiple classifiers to achieve a substantially improved prediction compared to a single classifier. Combining the results of multiple models can produce a statistically

significant and biologically relevant model for class prediction.

A major advantage of LORENS over CERP, which is tree-based, or other aggregation methods is that the base classifier is a well-received traditional regression method for binary response variables. It is not as computer intensive as CERP, while it does not lose the ensemble accuracy for high-dimensional data.

Like CERP, multiple ensembles are generated by randomly repartitioning the feature space and building base classifiers in order to achieve a further improvement. Majority voting is performed among these ensembles. In LORENS, a full logistic model is fit in each subset of the partition. However, different combinations of predictors in different ensembles will provide more information from a different partition in each ensemble.

The motivation for ensembles is to combine the outputs of many weak classifiers to produce a strong classifier (Breiman, 2001; Hastie et al., 2001). Kuncheva et al. (2003) illustrated the pattern of improving the accuracy in an ensemble by a majority voting. LORENS builds the ensemble using the CERP methodology which is designed for reducing the correlation among classifiers by random partitioning of the feature space. Ensemble error rate is most reduced in ensembles whose members make individual errors in a less correlated manner (Kuncheva et al., 2003; Hansen and Salamon, 1990).

LORENS is applied to the gene expression data on pediatric acute myeloid leukemia (AML) prognosis. Current chemotherapy enables a high percentage of patients with AML to enter complete remission, but a large number of them experience relapse with resistant disease (Yagi et al., 2003). Because of the wide heterogeneity of AML, predicting a patient's risk for treatment failure or relapse at the time of diagnosis is critical for an optimal treatment. We will investigate this gene expression data consisting of 53 AML pediatric patients (less than 15 years old) with an oligonucleotide microarray containing 12,566 probe sets. This data set is available at ftp://ftp.ncbi.nih.gov/pub/geo/DATA/SOFT/GDS/GDS1059.soft.gz. Among 53

pediatric patients with AML 21 of them are females and 32 are males. Patients with complete remission (CR) for more than 3 years are classified as good prognosis, while patients experienced induction failure or relapse within 1 year of the first CR are considered as poor prognosis (R: relapsed). The data set consists of 28 CR's and 25 R's. Using LORENS, we classify pediatric AML patients into CR and R in this paper.

We investigate the improvement of the prediction accuracy obtained from LORENS, and compare its performance with recently developed or commonly used ensemble methods including CERP, RF (Random Forest: Breiman, 2001), SVM (Vapnik, 1995) and boosting (Schapire, 1990; Freund and Schapire, 1996).

LORENS approach is comparable to other widely used ensembles in overall accuracy, with an excellent balance between sensitivity and specificity (balance) over SVM and Boosting according to our observation. This improvement of the balance was achieved by a decision threshold algorithmically obtained from the training phase. We implemented the LORENS algorithm in R. The software will be available upon request.

2 Methods

2.1 LORENS (Logistic Regression Ensembles)

Based on the CERP algorithm, we developed LORENS by using logistic regression models as base classifiers to classify pediatric AML prognosis. The goal of this study is, by combining patients' outcomes from a widely used logistic regression model, to achieve a classifier which is comparable to complex aggregation methods in terms of the prediction accuracy.

Let Θ be the space of the predictors. In order to minimize the correlation among the ensemble of classifiers, Θ is randomly partitioned into K subspaces $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_K)$ with roughly equal sizes. Since the subspaces are randomly chosen from the same dis-

tribution, we assume that there is no bias in selection of the genomic variables in each subspace. In each of these subspaces, we fit a full logistic regression model without a gene selection. Due to the randomness, we expect nearly equal probability of the classification error among the K classifiers and a similar improvement of the prediction accuracy.

LORENS combines the results of these multiple logistic regression models to achieve an improved accuracy of classifying a patient's outcome by taking average of the predicted values within an ensemble. The predicted values from all the base classifiers (logistic regression models) in an ensemble are averaged and classified as either 0 or 1 using a threshold on this average. Although the majority voting and averaging methods are similar, the latter gave slightly better prediction accuracy for LORENS in this study. LORENS generates multiple ensembles with different random partitions, and conduct a majority voting of individual ensembles to further improve the prediction accuracy. The multiple ensembles contribute to a further moderate gain in overall accuracy. Usually the gain is negligible when the number of ensembles exceeds ten. To avoid a tie in majority vote, eleven ensembles were used in this study.

In order to improve a balance between sensitivity and specificity, LORENS itself searches an optimal decision threshold for classification in the base classifiers of LORENS. We let r be the proportion of the positive responses in a data set. A threshold of r tends to give a better balance, and a threshold of $1 - r$ results in the highest accuracy (Chen et al., 2006). While a threshold of $1 - r$ tends to yield the highest prediction accuracy, it worsens the balance by predicting more samples to the majority class. To avoid sacrificing the accuracy and balance, we consider a threshold between 0.5 and r. We applied the same approach to RF when we compared the results in Section 3.

To search for the optimal threshold of LORENS in the learning phase, a nested 10-fold CV (cross validation) is performed in each learning set L_i, $i = 1, \ldots, 10$, of a 10-fold CV as follows: Within L_i, we use a finite grid with an increment of 0.02

between 0.5 and r (or between r and 0.5).

1. By applying each of the thresholds ts_j, $ts_j = .50, .52, \ldots, r$ (or $ts_j = r, r + .02, \ldots, .48, .50$), conduct the following nested 10-fold CV: Construct a LORENS classifier with one ensemble in each of the learning samples $L_{i(1)}, \ldots, L_{i(10)}$ and evaluate the accuracy using the corresponding test samples $T_{i(1)}, \ldots, T_{i(10)}$ using ts_j.

2. Choose a threshold with the highest prediction accuracy from part 1, say ts_i.

3. Apply ts_i to the test sample T_i corresponding to L_i.

Only one ensemble is used in this nested CV because of the tendency that the optimal threshold for LORENS is similar for one or multiple ensembles. This threshold is applied to the average response from the base classifiers within each ensemble.

The partition size is also determined in the training phase via 3-fold CV. Let n be the sample size of the data. In each learning set of a 3-fold CV, we first partition the predictor space such that a subspace has around $n/2$ predictors, build a LORENS model, and calculate the accuracy in each subspace. In this process, LORENS attempts $n/3, n/4, \ldots, n/10$ and $n/12$ for the size of each subspace. The partition size resulting in the highest overall accuracy is chosen among these. Thus n/i will be chosen for some integer i, $i = 2, \ldots, 10$ or 12. The second step is to search the optimal size of the subspaces by a dual binary search method between $n/(i-1)$ and n/i, and between n/i and $n/(i+1)$ based on the overall accuracy. From this, the one with higher overall accuracy between the two candidates is chosen in the learning phase.

2.2 Comparison of LORENS with existing classification methods

The comparison is based on the average of 20 trials of 10-fold CV. For the most relevant comparison, we provide the best result we obtained for the data set along with

some results from available options for each classification method. Specific parameters used are given below.

- CERP: The LR-T CERP (Logistic Regression Tree CERP) software written in R is used for the comparison. A rigorous optimization of the parameters using cross-validation is used as discussed in Ahn et al., (2007).

- RF: The RF package (RandomForest) in R is used for the comparison. The optimal values of *ntree* (the number of trees), *mtry* (the number of predictors randomly chosen in each node of a tree) and *cutoff* (the threshold for decision) leading to the highest overall accuracy are searched in the training phase using a nested CV in the same way as in LORENS.

- SVM: The e1071 package in R is used. We examined linear kernel and radial basis function (RBF). We attempted fine tuning of parameters by searching optimal value of epsilon. The value of epsilon was changed from 0.1 to 10, but no significant improvement was observed. Threshold was not applicable because SVM produces the class value of either 0 or 1, and there is no probability or fitted value in prediction. Thus we used default parameters.

- Boosting: The LogitBoost in the R package is used in the comparison. Performance of LogitBoost appears to be consistently good among the four boosting approaches provided in the R boosting package. We performed 100 iterations of weighted voting.

3 Results

3.1 Analysis of gene expression data on pediatric AML prognosis

We evaluated the performance of LORENS in classifying pediatric AML patients into CR (complete remission) and R (relapsed) groups from the gene expression data on AML discussed in Section 1. The comparison of LORENS with other methods were conducted by averaging the results from 20 trials of 10-fold CV in order to achieve a stable result.

We compared the prediction accuracy of LORENS with other widely used classification methods. Table 1 provides the accuracy of each method, with sensitivity (rate of correctly classifying into poor prognosis) and specificity (rate of correctly classifying into good prognosis), based on the average of 20 runs of 10-fold CV.

The accuracies of the methods were comparable except for SVM RBF. SVM RBF showed a slightly lower overall accuracy than the other methods. Although the data show only a slight imbalance (28 CR versus 25 R), SVM showed poor balance of sensitivity and specificity compared to the other classification methods. LORENS showed a nice balance. RF had a poor balance without the cutoff option. Since the feature space is huge with 12,566 predictors, the tuning of parameters were computationally not feasible for RF. Thus we used the default parameters of RF for this data set. The ROC curve and AUC cannot be used for assessing the balance because LORENS searches the optimal threshold in the learning phase, thus the threshold is not fixed.

The threshold search in LORENS was successful in terms of balancing sensitivity and specificity. For illustration, we tried a fixed threshold of 0.5 without the threshold search performed in training phase. The accuracy was not significantly affected by this (0.602, sd 0.046), but the sensitivity significantly decreased to 0.430 (sd 0.069) and the specificity significantly increased to 0.755 (sd 0.045) from almost equal numbers by the proposed method of the threshold-search (sensitivity 0.61 and specificity 0.63) shown

in Table 1. We observed a similar result from CERP. We expect that the balance would be improved if we conduct a similar threshold search for SVM. However, it is not straightforward to conduct the threshold search for SVM, while it is clearly straightforward for LORENS. SVM only gives the classes, but not probabilities, thus threshold cannot be applied.

3.2 Simulation study

We report the results of a simulation to evaluate the proposed LORENS as well as to compare its performance with that of other classification methods.

3.2.1 Experiment 1

This simulation study consisted of two parts. In the first part, predictors were independently generated, and in the second part, predictors were generated to have correlation. We generated two data sets with 100 subjects and 1000 predictors, one for training and the other for testing. Fifty of these predictor variables were generated from two different normal distributions, and the remaining 950 predictor variables were generated from one normal distribution. The latter served as noise. Fifty variables were generated from N(1, 1) for cases and N(0, 1) for controls. For models M1 and M2, these variables were independently generated. For models M3 and M4, correlation was given to each pair of these variables. The upper-diagonal elements of the correlation matrix were generated from a uniform distribution between 0 and 0.8. This correlation structure was generated once before the simulation, and used for generating simulation data in both M3 and M4. For each of M3 and M4, we generated the correlation matrices until we obtained a positive definite matrix. The average pairwise correlation obtained in this study was 0.43 for M3 and 0.41 for M4. The remaining 950 variables were independently generated from N(0, 1). The case-control ratios were given as 50:50 for M1 and M3, and 30:70 for M2 and M4. One hundred data sets were generated from each of the four models.

For each simulation data, the model fit in the training set was tested to the test set for evaluating the performance of each classification method. The average of the accuracies from the 100 pairs of simulation data sets from each classification method is provided in Tables 2 through 5. Figure 1 depicts the average accuracy of each classification method for each of the four simulation models.

Table 2 and Figure 1 show that all the methods considered here obtained accuracy of 96% or higher for M1. Table 2 shows that these methods also controlled balance between sensitivity and specificity for M1. For M2, SVM RBF showed low accuracy. SVM linear kernel appears to be successful in balancing sensitivity and specificity. All the methods performed well except SVM RBF for both of these models. However, LORENS is substantially less computer intensive than CERP because the former uses the logistic regression models, while the latter is a tree-based ensemble.

The accuracies of the classification methods were lower for the models with correlated data (see Tables 4 and 5). SVM RBF showed a severe imbalance between sensitivity and specificity for both models with unbalanced data (see Tables 3 and 5).

Table 6 compares the performance of LORENS between the fixed threshold and the threshold obtained in the learning phase. When we fix the threshold at 0.5 without the threshold search for LORENS, the accuracy and balance do not significantly change for M1 nd M3. However, the accuracy is substantially lower for M2 (0.96 to 0.70) and M4 (0.81 to 0.72). The balance is extremely worsen for M2 (0.00 and 1.00) and M4 (0.06 and 1.00). As addressed in Section 3.1, the procedure of searching an optimal threshold is essential in LORENS specifically when the data are unbalanced as shown in Table 6.

3.2.2 Experiment 2

In this simulation experiment, we generated data with independent predictors. We generated the data with the same size as in Section 3.2.1 with one strong predictor. One predictor variable was generated from two different normal distributions: N(0, 4)

and N(4, 4), and the remaining 999 predictor variables were generated from N(0, 4). Two models were considered. The case-control ratios were given as 50:50 for M5 and 30:70 for M6. One hundred data sets were generated from each of the two models.

Table 7 shows the result for M5 and Table 8 shows the result for M6. For M5, RF and LogitBoost gave almost perfect predictions, while the other methods yielded approximately 65% accuracy. The accuracy of LORENS was low because the significant predictor is available in only one subspace due to the random partition. In RF, variables are randomly chosen at each node of a tree, thus the significant predictor is included in many trees in the ensemble. For M6, no single model performed significantly better than the other models. Sensitivity and specificity were extremely unbalanced for all the models.

4 Conclusion and discussion

We have investigated an ensemble-based classifier with logistic regression models on each of the subsets in a random partition of the parameter space. LORENS has the following advantages compared to other well-known statistical classification methods.

- The variable selection can be computer-intensive for a high-dimensional data. Logistic regression models can be used for a high-dimensional data with an improved performance without variable pre-selection. A comparison between the performance of a single logistic regression with variable selection and LORENS can be done in a future work.

- LORENS is based on a widely used standard regression model for binary responses. Thus it is substantially less computer-intensive than tree-based CERP methods.

- It takes advantage of the CERP methods including low correlation among base classifiers by random partitioning of the feature space.

In a logistic regression model, the balance of sensitivity and specificity relied highly on the threshold of classification. The optimal thresholds from cross validation based on learning sets were searched in the LORENS algorithm. The balance was also significantly improved by the threshold search in RF. The balance of LORENS and RF were significantly improved compared to other aggregation methods such as SVM for unbalanced data sets. LORENS and RF showed consistently high classification accuracy when it was compared to other methods with the real and simulated data used in this study. A drawback on LORENS is that it does not perform well when there are few significant predictors in a high-dimensional feature space due to the random partition. RF and LogitBoost performs very well when for these types of data.

We applied the proposed method to gene expression data on pediatric AML patients. According to our study, LORENS was comparable to other widely used classification methods in predicting patient's risk for treatment failure or relapse at the time of diagnosis. We classified the AML patients to R and CR groups in this paper. Alternatively, we can obtain the probability using the average response of the ensembles. The statistical classification methods to predict pediatric AML prognosis introduced in this study are expected to play a critical role in developing safer and more effective therapies that replace one-size-fit-all drugs with treatments that focus on specific patient needs.

Acknowledgment

Hongshik Ahn's research was partially supported by the Faculty Research Participation Program at the NCTR administered by the Oak Ridge Institute for Science and Education through an interagency agreement between USDOE and USFDA. Hojin Moon's research was partially supported by the SCAC Award from CSULB.

References

1. Ahn, H., Moon, H., Fazzari, M. J., Lim, N., Chen, J. J., Kodell, R. L. (2007). Classification by ensembles from random partitions of high-dimensional data. *Computational Statistics and Data Analysis,* **51,** 6166-6179.

2. Breiman, L. (2001). Random Forest. *Machine Learning,* **45,** 5-32.

3. Chen, J. J., Tsai, C. A., Moon, H., Ahn, H., Chen, C. H. (2006). Decision threshold adjustment in class prediction. *SAR & QSAR in Environmental Research,* **17,** 337-351.

4. Freund, Y., Schapire, R. (1996). Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference,* 148-156.

5. Hansen, L. K., Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **12,** 993-1001.

6. Hastie, T., Tibshirani, R., Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* New York: Springer Verlag.

7. Kuncheva, L. I., Whitaker, C. J., Shipp, C. A., Duin, R. P. W. (2003). Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis and Application,* **6,** 22-31.

8. Vapnik, V. (1995). *The Nature of Statistical Learning Theory.* New York: Springer Verlag.

9. Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning,* **51,** 197-227.

10. Yagi, T., Morimoto, A., Eguchi, M., Hibi, S., Sako, M., Ishii, E., Mizutani, S., Imashuku, S., Ohki, M., Ichikawa, H. (2003). Identification of a gene expression signature associated with pediatric AML prognosis, *Blood,* **102,** 1849-1956.

Table 1: Accuracy (sd in parentheses) of classification methods for the gene expression data on pediatric AML prognosis. Twenty trials of 10-fold CV were performed for each method.

Method	Overall	Sensitivity	Specificity
LORENS[a]	.62 (.04)	.61 (.05)	.63 (.04)
CERP[b]	.63 (.04)	.62 (.04)	.65 (.04)
RF[c]	.62 (.05)	.49 (.09)	.73 (.04)
SVM RBF[d]	.55 (.05)	.39 (.05)	.68 (.06)
SVM Lin. Kernel	.58 (.03)	.44 (.05)	.71 (.04)
LogitBoost	.60 (.04)	.53 (.07)	.65 (.06)

[a] average partition size determined in the learning phase: 613
[b] LR-T CERP, average partition size determined in the learning phase: 598
[c] number of trees in each forest: 400; number of predictors: default (floor$[m^{1/2}]$)
[d] radial basis function (default option for the SVM function in the R package e1071)

Table 2: Accuracy (sd in parentheses) of classification methods for the simulation data from M1. One hundred pairs of training and test sets are generated.

Method	Overall	Sensitivity	Specificity
LORENS[a]	.99 (.01)	.99 (.01)	.99 (.01)
CERP[b]	.96 (.04)	.96 (.05)	.96 (.04)
RF[c]	.98 (.03)	.98 (.02)	.98 (.04)
SVM RBF[d]	.99 (.01)	.99 (.01)	.99 (.02)
SVM Lin. Kernel	.98 (.01)	.99 (.02)	.98 (.02)
LogitBoost	.97 (.02)	.97 (.03)	.98 (.03)

[a] average partition size: 60.4; average threshold: 0.5, obtained in the learning phase
[b] LR-T CERP, average partition size: 63.3; average threshold: 0.5, obtained in the learning phase
[c] average number of trees: 670; average number of predictors 54.1; average cutoff: 0.5, obtained in the learning phase
[d] radial basis function

Table 3: Accuracy (sd in parentheses) of classification methods for the simulation data from M2. One hundred pairs of training and test sets are generated.

Method	Overall	Sensitivity	Specificity
LORENS[a]	.96 (.06)	.97 (.05)	.95 (.09)
CERP[b]	.92 (.09)	.95 (.08)	.91 (.14)
RF[c]	.99 (.02)	.99 (.03)	.99 (.02)
SVM RBF[d]	.70 (.00)	.00 (.00)	1.00 (.00)
SVM Lin. kernel	.97 (.02)	.89 (.06)	1.00 (.00)
LogitBoost	.97 (.02)	.93 (.05)	.99 (.01)

[a] average partition size: 66.9; average threshold: 0.33, determined in the learning phase
[b] LR-T CERP, average partition: 63.5, average threshold: 0.33, determined in the learning phase
[c] average number of trees: 805; average number of predictors: 56.1; average cutoff: .35, determined in the learning phase
[d] radial basis function

Table 4: Accuracy (sd in parentheses) of classification methods for the simulation data from M3. One hundred pairs of training and test sets are generated.

Method	Overall	Sensitivity	Specificity
LORENS[a]	.80 (.04)	.80 (.07)	.81 (.07)
CERP[b]	.79 (.05)	.79 (.06)	.79 (.07)
RF[c]	.80 (.04)	.80 (.07)	.81 (.07)
SVM RBF[d]	.80 (.04)	.80 (.06)	.80 (.06)
SVM Lin. Kernel	.77 (.04)	.77 (.07)	.78 (.07)
LogitBoost	.75 (.05)	.74 (.07)	.76 (.07)

[a] average partition size: 66.1; average threshold: 0.5, determined in the learning phase
[b] LR-T CERP, average partition size: 57.1; average threshold: 0.5, determined in the learning phase
[c] average number of trees: 377; average number of predictors 47.9; average cutoff: 0.5, determined in the learning phase
[d] radial basis function

Table 5: Accuracy (sd in parentheses) of classification methods for the simulation data from M4. One hundred pairs of training and test sets are generated.

Method	Overall	Sensitivity	Specificity
LORENS[a]	.81 (.05)	.70 (.15)	.86 (.10)
CERP[b]	.81 (.04)	.65 (.16)	.87 (.09)
RF[c]	.83 (.03)	.70 (.11)	.88 (.04)
SVM RBF[d]	.70 (.01)	.01 (.02)	1.00 (.00)
SVM Lin. Kernel	.80 (.04)	.52 (.09)	.92 (.03)
LogitBoost	.78 (.04)	.51 (.11)	.89 (.04)

[a] average partition size: 65.7; average threshold: 0.35, determined in the learning phase
[b] LR-T CERP, average partition size: 69.9; average threshold: 0.38, determined in the learning phase
[c] average number of trees: 688; average number of predictors: 58.9; average cutoff: .38, determined in the learning phase
[d] radial basis function

Table 6: Comparison of the accuracies from Tables 2 to 5 with the accuracies obtained using a fixed threshold of 0.5 for LORENS for the same simulation data sets.

Model	Threshold	Overall	Sensitivity	Specificity
M1	Searched[a]	.99 (.01)	.99 (.01)	.99 (.01)
	Fixed at 0.5	.99 (.01)	.99 (.02)	.99 (.01)
M2	Searched	.96 (.06)	.97 (.05)	.95 (.09)
	Fixed at 0.5	.70 (.00)	.00 (.00)	1.00 (.00)
M3	Searched	.80 (.04)	.80 (.07)	.81 (.07)
	Fixed at 0.5	.80 (.04)	.80 (.06)	.80 (.07)
M4	Searched	.81 (.05)	.70 (.15)	.86 (.10)
	Fixed at 0.5	.74 (.04)	.16 (.15)	1.00 (.00)

[a] searched in the learning phase

Table 7: Accuracy (sd in parentheses) of classification methods for the simulation data with one strong predictor from M5 (balanced). One hundred pairs of training and test sets are generated.

Method	Model	Overall	Sensitivity	Specificity
LORENS[a]	.61 (.06)	.61 (.10)	.62 (.10)	
CERP[b]	.61 (.06)	.60 (.10)	.61 (.10)	
RF[c]	.98 (.02)	.98 (.03)	.97 (.03)	
SVM RBF[d]	.60 (.06)	.59 (.11)	.61 (.10)	
SVM Lin. Kernel	.60 (.06)	.60 (.10)	.61 (.09)	
LogitBoost	.97 (.02)	.97 (.03)	.98 (.02)	

[a] average partition size: 67.3, average threshold: 0.5, determined in the learning phase
[b] LR-T CERP, average partition size: 63.1, average threshold: 0.5, determined in the learning phase
[c] average number of trees: 627; average number of predictors 95.6; average cutoff: 0.5, determined in the learning phase
[d] radial basis function

Table 8: Accuracy (sd in parentheses) of classification methods for the simulation data with one strong predictor from M6 (unbalanced). One hundred pairs of training and test sets are generated.

Method	Model	Overall	Sensitivity	Specificity
LORENS[a]	.68 (.06)	.05 (.14)	.94 (.13)	
CERP[b]	.67 (.05)	.11 (.18)	.91 (.14)	
RF[c]	.69 (.03)	.05 (.06)	.96 (.05)	
SVM RBF[d]	.70 (.00)	.00 (.00)	1.00 (.00)	
SVM Lin. Kernel	.67 (.03)	.11 (.06)	.91 (.03)	
LogitBoost	.66 (.03)	.22 (.10)	.85 (.04)	

[a] average partition size: 67.3, average threshold: 0.5, determined in the learning phase
[b] LR-T CERP, average partition size: 63.1, average threshold: 0.5, determined in the learning phase
[c] average number of trees: 627; average number of predictors 95.6; average cutoff: 0.5, determined in the learning phase
[d] radial basis function

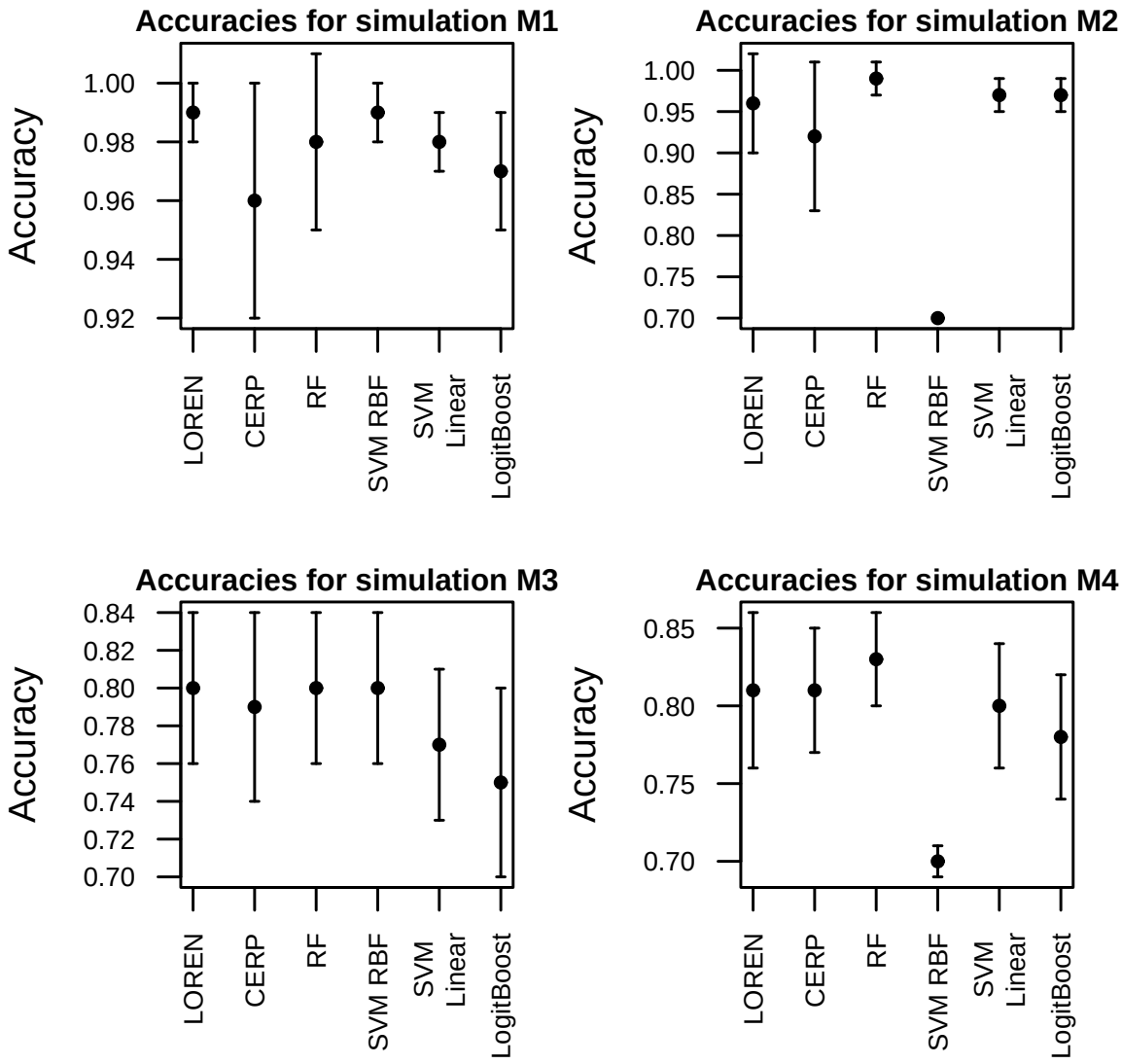


Figure 1: Comparison of accuracies (with 1-sd bars) of classification methods for each of the four simulation models given in Tables 2 through 5.