CLASSIFICATION OF MULTIPLE TIME-SERIES VIA BOOSTING

Patrick L. Harrington Jr.[*(1,2)], *Arvind Rao*[†(3)], *and Alfred O., Hero III.*[‡(1,2,4)]

(1) Bioinformatics Graduate Program, University of Michigan, Ann Arbor, MI 48109-2218, USA
(2) Department of Statistics, University of Michigan, Ann Arbor, MI 48109-1107, USA
(3) Lane Center for Computational Biology, Carnegie Mellon University, Pittsburgh, PA 15213, USA
(4) Department of EECS, University of Michigan, Ann Arbor, MI 48109-2122, USA

ABSTRACT

Much of modern machine learning and statistics research consists of extracting information from high-dimensional patterns. Often times, the large number of features that comprise this high-dimensional pattern are themselves vector valued, corresponding to sampled values in a time-series. Here, we present a classification methodology to accommodate multiple time-series using boosting. This method constructs an additive model by adaptively selecting basis functions consisting of a discriminating feature's full time-series. We present the necessary modifications to Fisher Linear Discriminant Analysis and Least-Squares, as base learners, to accommodate the weighted data in the proposed boosting procedure. We conclude by presenting the performance of our proposed method against a synthetic stochastic differential equation data set and a real world data set involving prediction of cancer patient susceptibility for a particular chemoradiotherapy.

***Index Terms*—** Boosting, Time-Series, Additive Models

1. INTRODUCTION

Modern scientific problems are increasingly confronted with high-dimensional data containing subtle patterns that can provide a wealth of information for explaining the variation between the classes that produced it. For example, in predicting breast cancer patient prognosis/survival, researchers derived a 70-dimensional molecular "signature" that was more predictive of future patient status than traditional clinical methods [1]. From this motivating example, properly predicting a patient's future disease status would allow for more tailored therapeutic options that minimize the cost associated with mistreatment and more importantly, maximize patient survival/recovery.

Increasingly often, these high-dimensional patterns, containing thousands of different features, have been observed at multiple time-points or are themselves multivariate. In the former case, a physician may monitor a patient's gene expression activity over a sequence of visits before determining the most appropriate therapy. Since microarray gene expression measurements consist of a snapshot of thousands of different genes, we are presented with a high-dimensional data set of features over different observed time points [2]. Another example is in surveillance, where a satellite may acquire many different images of a particular ground surface or potential enemy target taken under different spectral bands and the goal is try classify these surfaces or objects [3]. In addition to performing classification, it also of interest in many problems to extract which of these features, e.g., genes or images, are important in capturing the variation between the classes and what subspace is responsible for this variation, e.g., time-points or regions of an image.

In this report, each time-series or multivariate feature will be represented by a vector. The modified boosting algorithm will selects a basis function consisting of the most discriminating feature at the m^{th} boosting iteration with respect to one of the two modified base learners: Fisher Linear Discriminant Analysis and Least-Squares. The resulting classifier will consist of different features where each member to the additive model attempts to linearly partition this selected feature's vector-space. Our discussion will be restricted to the Discrete AdaBoost algorithm and the extension of the proposed base learners to weighted data is presented. We present numerical results of the proposed method on both a synthetic data set of stochastic differential equations and a real data set involving patients with rectal cancer. We conclude with a discussion of future work.

2. METHODS

2.1. Additive Models

Additive models for regression and classification have received a significant amount of attention in the statistics and machine learning community as powerful tools for explain-

[*]plhjr@umich.edu. Thanks to NIH Training Grant No. 1 T32 GM070449-01A1

[†]ukarvind@andrew.cmu.edu. Thanks to Rackham Predoctoral Fellowship, University of Michigan

[‡]hero@umich.edu. Thanks to NSF Grant No. CCR-0325571 for Funding

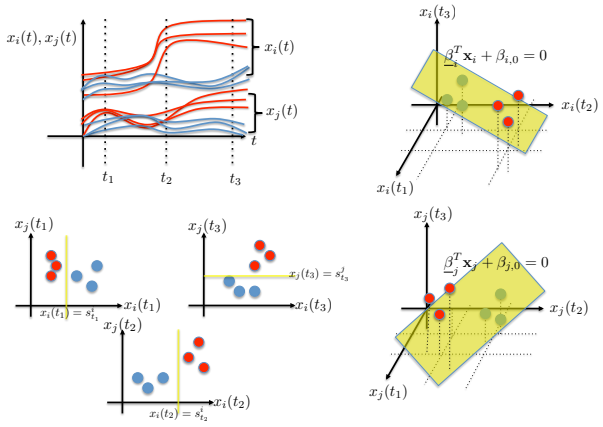


Fig. 1. Upper Left: Time-series of features i and j sampled at 3 time points. Lower Left: Traditional approach isolates individual time-slices and produces three distinct learners (shown are decision stumps that partition space based on scalar split at m^{th} boosting iteration - yellow lines) Right:For feature i and j, their time-series is a 3-dim vector and a linear classifier is used to partition this space.

ing the data [4, 5, 6, 7, 8, 9]. In classification, the additive model may be represented as a basis function expansion of individual learners

$$F_M(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m f_m(\mathbf{x}; \gamma_m) \tag{1}$$

with expansion coefficient α_m, base learner $f_m(\mathbf{x}; \gamma_m)$, and basis function dictionary parameter γ_m. Here, $\mathbf{x} \in \mathbb{R}^p$, is the p-dimensional feature vector, e.g., pixels in an image or transcripts on a microarray platform. The canonical statistical method for constructing (1) is to adopt a loss function, e.g., quadratic, binomial deviance, hinge, etc, and proceed with the estimation. However, since the basis functions are not fixed *a priori*, obtaining (1) requires a combinatorial search over all possible basis function dictionary parameters $\{\gamma_m\}_{m=1}^M$, which quickly becomes infeasible with the size of the dictionary and expansion number M. Thus, approximate methods must be used to obtain (1) [4, 5, 6, 7]. For this report, we will focus on a greedy forward stage approach known as boosting [4, 6], and restrict our attention to the exponential loss $L(y, f) = \exp(-yf)$ under training data $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ with binary response $y \in \{-1, +1\}$.

2.2. Boosting

Boosting is notably one of the most powerful off-the shelf classification algorithms [10] and constructs (1) via a greedy forward stage method (under an exponential loss) :

$$\begin{aligned}\{\alpha_m, f_m\} &= \arg\min_{\alpha,f} \sum_{j=1}^{n} \exp\left(-y_j \left(F_{m-1}(\mathbf{x}_j + \alpha f(\mathbf{x}_j))\right)\right) \\ &= \arg\min_{\alpha,f} \sum_{j=1}^{n} w_j \exp\left(-\alpha y_j f(\mathbf{x}_j)\right) \end{aligned} \tag{2}$$

for $f \in \{-1, +1\}$. There exists many boosting algorithms, accounting for different loss functions and constraints on f, and all can be extended to the methods proposed in this report. However, we will limit our discussion to the Discrete AdaBoost algorithm and delay the generalization to other loss functions for future work [6, 4, 10]. The solution to (2.2), for all m, is given below in Algorithm 1:

Algorithm 1: Discrete AdaBoost

1. Initialize the observations weights $w_j^{(0)} = 1/n, j = 1, \ldots, n$.
2. For $m = 1$ to M
   (a) Fit a classifier $f_m(\mathbf{x})$ to the training data using weights $w_j^{(m)}$:
   (b) $f_m(\mathbf{x}) = \arg\min_{f \in \mathcal{F}} \hat{\mathbb{P}}_w \left(y \neq f(\mathbf{x})\right)$
   (c) Note: $\hat{\mathbb{P}}_w \left(y \neq f_m(\mathbf{x})\right) = \sum_{j=1}^{n} w_j^{(m)} \mathbb{I}_{\{y_j \neq f_m(\mathbf{x}_j)\}}$ with $\sum_{j=1}^{n} w_j^{(m)} = 1$
   (d) If $\hat{\mathbb{P}}_w \left(y \neq f_m(\mathbf{x})\right) > 1/2$, flip polarity of $f_m(\mathbf{x})$
   (e) Compute $\alpha_m = \log\left((1 - \text{err}_m)/\text{err}_m\right)$ with $\text{err}_m = \hat{\mathbb{P}}_{\mathbf{w}} \left(y \neq f_m(\mathbf{x})\right)$
   (f) Set $w_j^{(m)} = w_j^{(m-1)} \exp\left[\alpha_m \mathbb{I}_{\{y_j \neq f_m(\mathbf{x}_j)\}}\right], j = 1, \ldots, n$
3. Output $F_M(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m f_m(\mathbf{x})$ with $\hat{y} = \text{sign}\left[F_M(\mathbf{x})\right]$.

It was shown in [4] that under the exponential loss, the final classifier F_M in (1) is estimating half the log-odds of the posterior probability of class assignment. Thus, the posterior probability of class labels can be recovered from the final F_M.

2.3. Extension to Multiple Time-Series

Often times, the signals are high dimensional which jointly evolve according to some stochastic process, sampled at discrete time points. Since distribution information regarding is usually unavailable, we must proceed using training data. The goal is to extract predictive patterns on a subset of features that explain the variation between multiple class labels. From the supervised learning perspective, we no longer have a p-dimensional vector, $\mathbf{x}$ but a set of p random variables corresponding to the sampled values for each of the p features, i.e., $\mathbf{x}_i = [x_i(t_1), \ldots, x_i(t_d)]^T$ with $x_i(t_k)$ corresponding to the sampled value of feature i at time t_k (dependent on class label y). Thus, our training data is of the form $\mathcal{D} = \{\{\mathbf{X}_i\}_{i=1}^p, \mathbf{y}\}$, with n by d design matrix for the i^{th} feature $\mathbf{X}_i$ and response column vector $\mathbf{y} = [y_1, \ldots, y_n]^T$. The goal is to construct (1)

according to Algorithm 1 where the m^{th} basis function is a function of one of the p features with basis function dictionary parameter $i_m = \gamma_m \in \{1, \ldots, p\}$:

$$\{i_m, f_m\} = \text{argmin}_{i \in \{1,\ldots,p\}} \left\{ \text{argmin}_{f\{-1,+1\}} \hat{\mathbb{P}}_w(y \neq f(\mathbf{x}_i)) \right\}. \tag{3}$$

Here, we will restrict our class of basis functions f to be the class of linear classifiers that, for the i^{th} feature, linearly partition the d-dimensional vector space corresponding to the d sampled values of process $x_i(t)$. In other words, at the m^{th} boosting iteration, the decision region for positively assign points is given by $\mathcal{X}^+_{i_m} = \{\mathbf{x}_{i_m} : \underline{\beta}^T_{i_m} \mathbf{x}_{i_m} + \beta_{i_m,0} > 0\}$ whereas the decision region for the negative class assignments is $\mathcal{X}^-_{i_m} = \{\mathbf{x}_{i_m} : \underline{\beta}^T_{i_m} \mathbf{x}_{i_m} + \beta_{i_m,0} < 0\}$ (see Fig.1). In this report, we will restrict our attention to the following linear classifiers: Fisher Linear Discriminant Analysis and Least Squares. We present the necessary modifications to both methods to accommodate the weighted data associated with the boosting procedure below.

2.3.1. Weighted Fisher Linear Discriminant Analysis

Fisher posed the following problem for linearly discriminating different classes of data in the following way: "Find the linear combination of original data such that the between-class variance is maximized relative to the within-class variance". This reduces to finding the vector $\underline{\beta}_i$, for the i^{th} feature, which maximizes the following objective function (subscript i removed for clarity):

$$\max_{\underline{\beta}} \frac{\underline{\beta}^T \mathbf{B} \underline{\beta}}{\underline{\beta}^T \mathbf{W} \underline{\beta}} \tag{4}$$

Since the objective function is invariant to rescaling of $\underline{\beta}$, we can reformulate this as the following constrained optimization problem:

$$\max_{\underline{\beta}} \frac{1}{2} \underline{\beta}^T \mathbf{B} \underline{\beta} \text{ s.t } \underline{\beta}^T \mathbf{W} \underline{\beta} = 1 \tag{5}$$

Here, the between and within class covariance matrices, **B** and **W**, respectively. Given the constrained optimization problem in (5), we can write the Lagrangian,

$$\mathcal{L}_P = \frac{1}{2} \underline{\beta}^T \mathbf{B} \underline{\beta} - \frac{1}{2} \lambda (\underline{\beta}^T \mathbf{W} \underline{\beta} - 1). \tag{6}$$

Differentiating with respect to $\underline{\beta}$ and equating to zero, we arrive at the following generalized eigenvalue equation:

$$\mathbf{W}^{-1} \mathbf{B} \underline{\beta} = \lambda \underline{\beta}. \tag{7}$$

The problem now reduces to solving an eigen-decomposition of $\mathbf{W}^{-1}\mathbf{B}$ and selecting the eigenvector corresponding to largest eigenvalue, the maximizer of (4). However, in the boosting framework, our training data have corresponding weights and our estimation of the common/class centroids and covariance matrices must account for these. The estimation of these parameters under potentially weighted data points for the i^{th} feature are given by:

Estimation of Parameters for Weighted Fisher LDA

1. $\bar{\mathbf{x}}_i = \sum_{j=1}^N w_j \mathbf{x}_i^j$
2. $\hat{\underline{\mu}}_i^k = \sum_{j:y_j=k} w_j \mathbf{x}_i^j / \sum_{j:y_j=k} w_j, \ k \in \{-1, +1\}$
3. $\mathbf{B}_i = \sum_k W_k (\hat{\underline{\mu}}_i^k - \bar{\mathbf{x}}_i)(\hat{\underline{\mu}}_i^k - \bar{\mathbf{x}}_i)^T, W_k = \sum_{j:y_j=k} w_j$
4. $\mathbf{W}_i = \sum_k \sum_{j:y_j=k} w_j (\mathbf{x}_i^j - \hat{\underline{\mu}}_i^k)(\mathbf{x}_i^j - \hat{\underline{\mu}}_i^k)^T$
5. Note: if $w_j = 1/n$ for all $j \in \{1, \ldots, n\}$, these reduce to the unweighted sample mean and sample covariance matrix.

Given these estimates, we can proceed with the eigen-decomposition of $\mathbf{W}_i^{-1}\mathbf{B}_i$ and obtain the normal vector, $\underline{\beta}_i$.

2.3.2. Weighted Least Squares

The second linear classification method explored throughout this paper is weighted least-squares of a binary response variable. Least-squares easily accommodates the boosting weights by assigning importance to the residuals that "matter the most":

$$\hat{\underline{\beta}}_i = \arg\min_{\underline{\beta}} \sum_{j=1}^n w_j \left(y_j - \underline{\beta}^T \mathbf{x}_i^j \right)^2 \tag{8}$$

or gathering all n terms into vector and matrix notation:

$$\hat{\underline{\beta}}_i = \arg\min_{\underline{\beta}} \left(\mathbf{y} - \mathbf{X}_i \underline{\beta} \right)^T \mathbf{W} \left(\mathbf{y} - \mathbf{X}_i \underline{\beta} \right) \tag{9}$$

with n by d design matrix $\mathbf{X}_i$ for the i^{th} feature, diagonal matrix $\mathbf{W} = \text{diag}(w_j)$, and the response column vector $\mathbf{y}$. Here, we have assumed that the intercept term β_0 has been absorbed into $\mathbf{y}$ (implies that $\mathbf{X}_i$ is centered). The normal vector $\underline{\beta}$ that minimizes (9) is given by:

$$\hat{\underline{\beta}}_i = \left[\mathbf{X}_i^T \mathbf{W} \mathbf{X}_i \right]^{-1} \mathbf{X}_i^T \mathbf{W} \mathbf{y}. \tag{10}$$

The m^{th} basis function, or base learner, for the i^{th} feature is then given by $f_m(\mathbf{x}_i) = \hat{\underline{\beta}}_i^T \mathbf{x}_i + \hat{\beta}_0$, producing a signed distance from some test point $\mathbf{x}_i$ to the linear decision boundary.

3. NUMERICAL RESULTS

3.1. Synthetic Data

Here we present the results of our proposed method on a synthetic time-series training data set, consisting of 100 different features, evolving over 100 time points (sampled at 20 evenly spaced time points). There are three different features that contain a signal (shown in Fig.2 as green, red, and blue) for class $y = -1$ while the rest are noise. All of the features evolve according to different stochastic differential

equations with common volatility but the three signal features having different non-zero drift terms, yielding the peaks shown in Fig.2. The 50 iid samples of the three signal features ($y = -1$), along with 50 more of these features with no drift term ($y = 1$), are both depicted in Fig.2.

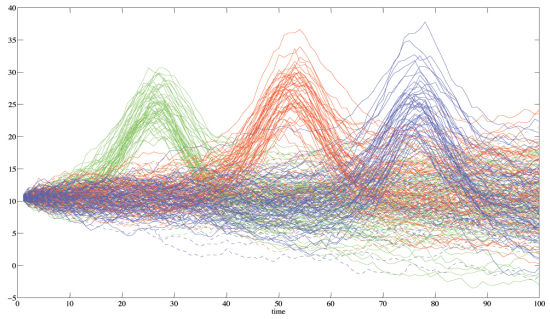


Fig. 2. Full synthetic time-series of 100 different features (only 3 signal features, solid lines, and corresponding noise shown, dashed lines), sampled every 5 time-points. The 100 different features are generated from different stochastic differential equations

Training the classifier on this 100 feature data set, we have extracted the three signal features as the *only* discriminating features in the data set (see Fig.3). The remaining noise features have importances of zero since they were not selected throughout the boosting procedure.

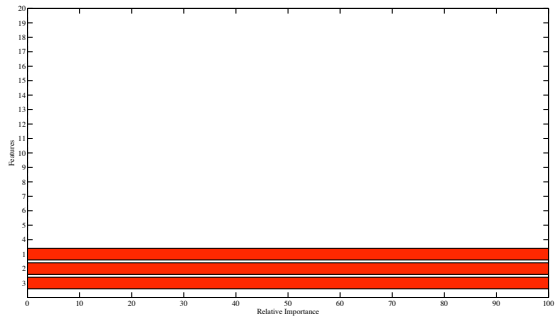


Fig. 3. Feature Importance: The boosting method selects only the three signal features (all equally "important") whereas the remaining noise features (not all listed) have importance of zero because they were not selected as discriminators in any of the M boosting iterations.

It is obvious from Fig.2 that the maximal variation between the two classes is captured at the respective peaks of the three signals shown (features green, red, and blue). Hence, we would expect that within this 20-dimensional space, there exists a subspace that is responsible for this variation. Both linear classifiers used within the boosting framework have penalized counterparts, which can extract a sparse normal vector $\underline{\beta}$ that determines this linear subspace based upon the non-zero elements of $\underline{\beta}$ [11, 12]. By varying the shrinkage factor s between 0 (total shrinkage) and 1 (no shrinkage) using weighted least-squares, formulated as LASSO [12], we extract the importance of the features, as depicted in Fig.3, and the discriminating linear subspace Fig.4. Based upon the first coefficient to appear non-zero under small s in Fig.4a, we see for feature 1 (green), the most discriminating time-point is $t = 25$, which corresponds to the sampled point closest to the peak in Fig.2. The same follows for features 2 and 3 which yield the most discriminating time points of $t = 50$ and $t = 75$, respectively. We see that the magnitude of all the normal vector coefficients for feature 3 (blue) are larger than those for features 2 and 3. This is explained because the noise term for all features is a Wiener process, which has variance linear in t.

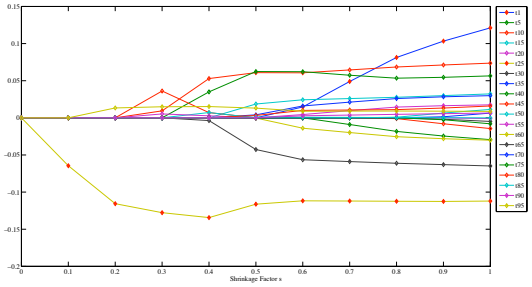
Plotting both the process of the three signal features and the absolute value of the normal vector coefficients, at a shrinkage factor of $s = 0.2$, we see that the mass of the coefficients is being assigned to the time-points which are most discriminating in the process (see Fig.5).

The method described throughout this paper allows for a simultaneous execution of three important tasks in a machine learning problem: classification/prediction, relative feature importance, and subspace importance. Here, the subspace corresponds to sampled time-points but can can be extended to any problem containing multiple random-vectors, e.g., satellite image data under different spectral bandwidths taken under a variety of conditions.

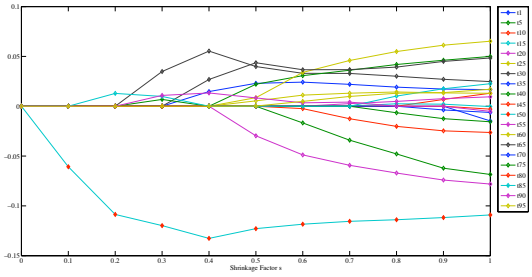
3.2. Cancer Data

In bioinformatics research, physicians are increasingly faced with developing personalized therapies for groups of patients who are susceptible to a particular treatment while avoiding mis-treatment of those who are not receptive to such treatments. Often times, the data is generated from a series of patient visits, thus forming a time-series, and the physician is interested in predicting whether a patient will respond desirably to a proposed treatment strategy. In [2], Luminex proteomic data and Affymetrix gene expression data was acquired from 36 rectal cancer patients at three distinct time points where the phenotypes have been labeled RCRG positive ($y = +1$ - good responsiveness) or RCRG negative ($y = -1$ - moderate/poor responsiveness). In this study, the researchers were interested in predicting the phenotype at the third time point given the previous two using least-squares support vector machines (LS-SVM) [13, 2].

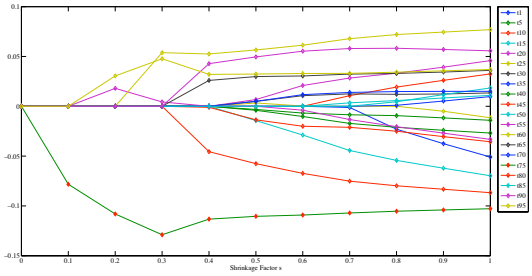
We present the generalized performance of the proposed boosting multiple time-series (BMTS), with Fisher LDA and LS as the base learners, compared with the approach taken in [2]. In both studies, the genes and proteins were rank-ordered in terms of their statistical significance using a rank-sum test, and the structure of the classifiers were constructed



(a) Shrunken Normal Vector Coefficients for Feature 1 (green)



(b) Shrunken Normal Vector Coefficients for Feature 2 (red)



(c) Shrunken Normal Vector Coefficients for Feature 3 (blue)

Fig. 4. Subspace Importance (time-points): Normal Vector Coefficients As a Function of Decreasing Degree of Shrinkage

using a subset of significant features. The generalized performance was assessed using a leave-one-out cross validation. While data fusion was an additional motivation in [2], simply forming a master kernel, which was a convex combination of kernels trained on individual time-points and data sources, we treated each data modality separately and present the results in Table 1. We see that our proposed method performs equally well against the LS-SVM method when using 5 proteins while the FLDA base learner outperforms the LS-SVM when 25 proteins were used. The authors in [2] inspected 1000 and 3000 genes obtaining significantly poorer performance than the BMTS approach using far fewer genes Table 1. We believe performance could be further enhanced by extending the basis functions used in BMTS to include quadratic classifiers, e.g., quadratic discriminant analysis, a consideration for future-work.

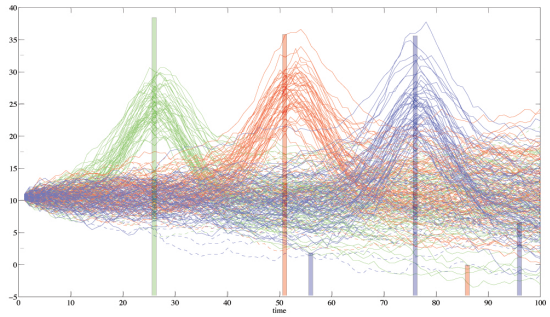


Fig. 5. Estimated Normal Vector Coefficients Capture the Maximal Variation Between Classes at Respective Peaks for $s = 0.2$

Table 1. Performance Results

Method	N_g	N_p	Sens.	Spec.	Accuracy
LS-SVM t_2	-	5	0.885	0.8	0.861
BMTS-FLDA	-	5	0.885	0.8	0.861
BMTS-LS	-	5	0.885	0.8	0.861
LS-SVM t_1 and t_2	-	25	0.808	0.7	0.778
BMTS-FLDA	-	25	0.923	0.6	0.833
BMTS-LS	-	25	0.846	0.6	0.778
BMTS-FLDA	5	-	0.885	0.8	0.861
BMTS-LS	5	-	0.923	0.7	0.861
BMTS-FLDA	25	-	0.846	0.8	0.833
BMTS-LS	25	-	0.846	0.7	0.801

4. DISCUSSION AND FUTURE WORK

While the work within this paper has been confined to the Discrete AdaBoost algorithm, the extension to multiple time-series can be generalized to accommodate other loss functions within the forward stage additive modeling framework. We have presented the necessary modifications to a restricted class of linear classifiers to accommodate the weights associated with boosting training data. Fisher LDA and Least-Squares are desirable linear base learners for our boosting procedure as they both can be formulated as penalized optimization problem, producing a shrunken β, corresponding to a discriminating linear subspace [11, 12]. The authors believe that support vector machines are a strong candidate for

base learners within BMTS given their natural ability to extend to accommodate nonlinear data, individual penalization of the slack variables (needed to handle different boosting weights), and sparse formulations, necessary for subspace selection [14, 15]. Since the sampled data points for each of the p features most likely lives on some lower-dimensional manifold, we will extend our proposed method to exploit such possible structure for classification and subspace selection. Given the relationship between boosting and additive models, our next step is to provide theoretical performance results for boosting multiple time-series under different loss functions. In many problem domains, especially bioinformatics, the notation of simultaneously performing classification/regression, feature selection, and subspace selection is important, and we believe our proposed BMTS method, which utilizes selection and regularization, we believe BMTS is an attempt to satisfy these requirements.

5. SUPPLEMENTAL MATERIAL

The MATLAB toolbox accompanying this report, Boosting Multiple Time-Series (BMTS), is available for download at http://www.umich.edu/∼plhjr/Software.html. Special thanks to Mark Schmidt for making his MATLAB lasso solver available for download at http://www.cs.ubc.ca/∼schmidtm/Software/lasso.html. The authors also thank Mark Kliger for making his Sparse Eigen-Methods toolbox available as well (embedded within the BMTS toolbox).

6. REFERENCES

[1] Laura J van 't Veer, Hongyue Dai, Marc J van de Vijver, Yudong D He, Augustinus A M Hart, Mao Mao, Hans L Peterse, Karin van der Kooy, Matthew J Marton, Anke T Witteveen, George J Schreiber, Ron M Kerkhoven, Chris Roberts, Peter S Linsley, Rene Bernards, and Stephen H Friend, "Gene expression profiling predicts clinical outcome of breast cancer.," *Nature*, vol. 415, no. 6871, pp. 530–536, 2002.

[2] Anneleen Daemen, Olivier Gevaert, Tijl De Bie, Annelies Debucquoy, Jean-Pascal Machiels, Bart De Moor, and Karin Haustermans, "Integrating microarray and proteomics data to predict the response on cetuximab in patients with rectal cancer.," *Pac Symp Biocomput*, pp. 166–177, 2008.

[3] A. O. Hero, D. Castenon, D. Cochran, and K. D. Kastella, *Foundations and applications of sensor management*, Springer, NY, 2007.

[4] J Friedman, T Hastie, and R Tibshirani, "Additive logistic regression: A statistical view of boosting," *ANNALS OF STATISTICS*, vol. 28, no. 2, pp. 337–374, APR 2000.

[5] Trevor Hastie and Robert Tibshirani, *Generalized Additive Models*, Chapman and Hall, 1990.

[6] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of the Thirteenth International Conference on Machine Learning (ICML)*, 1996, pp. 148–156.

[7] Robert E. Schapire, "Theoretical views of boosting and applications," in *Algorithmic Learning Theory, 10th International Conference, ALT '99, Tokyo, Japan, December 1999, Proceedings*. 1999, vol. 1720, pp. 13–25, Springer.

[8] Leo Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[9] Leo Breiman, "Random forests," *Machine Learning*, vol. V45, no. 1, pp. 5–32, October 2001.

[10] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning*, Springer, August 2001.

[11] Baback Moghaddam, Yair Weiss, and Shai Avidan, "Generalized spectral bounds for sparse lda," in *Proc. ICML*, 2006.

[12] R Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Series B-Methodological*, vol. 58, pp. 267–288, 1996.

[13] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

[14] Vladimir N. Vapnik, *The nature of statistical learning theory*, Springer-Verlag New York, Inc., 1995.

[15] Antoni B. Chan, Nuno Vasconcelos, and Gert R. G. Lanckriet, "Direct convex relaxations of sparse svm," in *ICML '07: Proceedings of the 24th international conference on Machine learning*, 2007, pp. 145–153.