# Finding predictive runs with LAPS

Suhrid Balakrishnan
Department of Computer Science,
Rutgers University, Piscataway, NJ 08854 USA
suhrid@cs.rutgers.edu

David Madigan
Department of Statistics,
Rutgers University, Piscataway, NJ 08854 USA
dmadigan@rutgers.edu

## Abstract

*We present an extension to the Lasso [9] for binary classification problems with ordered attributes. Inspired by the Fused Lasso [8] and the Group Lasso [10, 4] models, we aim to both discover and model runs (contiguous subgroups of the variables) that are highly predictive. We call the extended model LAPS (the Lasso with Attribute Partition Search). Such problems commonly arise in financial and medical domains, where predictors are time series variables, for example. This paper outlines the formulation of the problem, an algorithm to obtain the model coefficients and experiments showing applicability to practical problems of this type.*

## 1 Predictive Runs

We consider regression and classification problems where the predictor variables are ordered and naturally form groups. For example, in predicting whether or not a vaccinated animal survives an anthrax challenge, relevant attributes might include a toxin neutralization assay (TNA) measured at ten different time points (i.e., a group of ten predictor variables), a protective antigen assay, measured at 20 time points (i.e., a group of twenty predictor variables), and vaccine dilution (i.e., a group containing a single predictor variable). We believe that many regression and classification applications exhibit such structure and we describe several in what follows.

Standard modelling approaches that ignore the group structure or ordering can lead to models that provide good predictive performance but make little sense. For example, applying feature selection to the problem above might result in selecting TNA predictor variables corresponding to measurements at weeks 2, 8, and 48 and dropping the measurements at weeks, 4, 6, 12, 20, 32, 40, and 52. Similarly, feature selection might result in selecting values of other assays at seemingly arbitrary timepoints. Since the assay measurements are generally serially correlated, small perturbations to the training data often lead to the selection of a drastically different set of predictor variables. We argue that in many applications, it makes more sense to select (or omit) contiguous "runs" of predictor variables - for example, TNA measurements from week 2 through week 8.

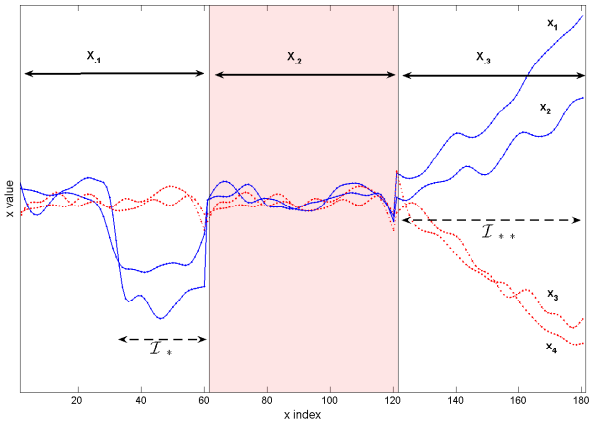


**Figure 1. Typical classification problem setup. Plotted are 4 examples, two each drawn from the two different classes (shown in red dotted and blue solid lines). Also shown in the figure are the 3 different groups (the differently shaded and delineated bands $\mathbf{x}_{.1}, \mathbf{x}_{.2}, \mathbf{x}_{.3}$) and potential locations for predictive runs.**

In this paper we focus on binary classification problems. For each example, given an input vector $\mathbf{x}_i = [\mathbf{x}_{i1}, \ldots, \mathbf{x}_{id}]$, we seek to predict the corresponding label $y_i \in \{-1, 1\}$. Each $\mathbf{x}_{ij}$ corresponds to a "group," $\mathbf{x}_{ig} = [x_{ig}^{(1)}, x_{ig}^{(2)}, \ldots, x_{ig}^{(T_g)}], T_g \geq 1$(the group length). Figure 1 shows a problem with three equal-size groups, $\mathbf{x}_{.1}, \mathbf{x}_{.2}$ and $\mathbf{x}_{.3}$, where the second half of the first group (variable indices $\sim$ 30—60, denoted by $\mathcal{I}_*$) has some discriminatory power with respect to the two classes (indicated by blue-solid and red-dotted lines), the second group has no dis-

criminatory power, and the entire third group (indices $\mathcal{I}_{**}$), is useful, with the possible exception of the first few values.

We restrict our attention to linear logistic regression models for interpretability, and we seek to develop a modelling approach that automatically identifies the sub-group indices, or runs, $\mathcal{I}_*$ and $\mathcal{I}_{**}$ and the corresponding regression parameters.

## 2 Modelling Predictive Runs

Our work builds on two recent extensions to the original "Lasso" $L_1$-regularized regression models of Tibshirani [9] [1]. The "fused Lasso" [8] addresses problems like ours where the variables are ordered (their development and experiments are described for inputs with one group, i.e., $\mathbf{x} = \mathbf{x}_{.1}$). The fused Lasso encourages contiguous subgroups of **identical** coefficients (the corresponding variables being highly correlated) to be non-zero together. They accomplish this through an additional $L_1$ penalty (besides the regular Lasso regularization term) on the differences of successive coefficient values ($\beta_k - \beta_{k+1}$ terms). In a sense, what we propose below is a "soft" version of the fused lasso.

Our work is more closely related to another Lasso extension, the "group Lasso" [10, 4]. Here the emphasis is on adapting the Lasso sparsity to **sets of predictors**. In particular, the group lasso either selects or omits entire groups of variables, where the data analyst pre-specifies what attributes form the groups. An elegant result of this formulation is that it reduces to the Lasso when all the variable sets are of size one.

Here, we propose a data-driven approach to identify runs (or contiguous subgroups) of model coefficients, that are similar (like a soft fused Lasso) and that will be selected together (have non-zero model coefficients en-block, like the group Lasso). The challenge is that we neither know the within-group run structure of the attributes, nor the amount of similarity within runs beforehand. The following sections outline our approach to these problems, which essentially consists of modifying the group Lasso penalty to potentially include similarity between coefficients and searching over group partitions into runs. We call this approach LAPS (the Lasso with Attribute Partition Search).

### 2.1 The LAPS model

Given hyper-parameters $\lambda$ (a regularization parameter) and $k$ (a parameter governing serial correlation of run coefficients), LAPS finds logistic regression coefficients $\boldsymbol{\beta}$ and the run structure $\mathcal{I}$ such that:

$$\underset{\boldsymbol{\beta},\mathcal{I}}{\operatorname{argmin}}\, nll(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{J} s_j \left\| \boldsymbol{\beta}_{\mathcal{I}_j} \right\|_{K_j}, \qquad (1)$$

where $nll(\boldsymbol{\beta}) = -\sum_{i=1}^{t} \log \Phi(-y_i \boldsymbol{\beta}^T \mathbf{x}_i)$, is the negative log-likelihood involving the logistic link function $\Phi(z) = \frac{e^z}{1+e^z}$. The training data $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^{t}$ comprises $t$ labeled examples where the input examples, as before, are $\mathbf{x}_i = [\mathbf{x}_{i1}, \ldots, \mathbf{x}_{id}]$ and can be thought of compactly as a single $p$-dimensional vector (as presented in the introduction. Thus $p$ = the total number of attributes = sum of the lengths of the $d$ groups = $\sum_{g=1}^{d} T_g$). The run structure (or partition structure) $\mathcal{I}$ comprises the set of run indices $\mathcal{I}_j$, $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_J\}$. These run indices, in turn, form a disjoint partition that covers the entire attribute index range (all groups), thus $\cup_{j=1}^{J} \mathcal{I}_j = 1 \ldots p$, and $\mathcal{I}_u \cap \mathcal{I}_v = 0, \forall u \neq v$. Additionally we impose the requirement that run indices respect all group boundaries (that is, runs never cross the boundaries between the various $\mathbf{x}_{.j}$ for different $j$s). The $K_j$ matrices are positive definite matrices parameterized by a single scalar $k$ (subsection 2.2 has details). The regularization term involves the $K_j$ matrix norms[2] of the run coefficients (for a vector $\mathbf{z}$ and a matrix $A$, $\|\mathbf{z}\|_A = (\mathbf{z}^T A \mathbf{z})^{0.5}$). Finally, the $s_j = \sqrt{0.5(|\mathcal{I}_j| + 1)}$, are scalars factors that "normalize" the prior $\boldsymbol{\beta}$ variance (more about this also in subsection 2.2).

Figure 2 shows the resulting LAPS model applied to a small simulated example with two groups, where one entire group is discriminative (the shaded/right group of indices). Whereas the Lasso results are unsatisfactory, as it finds only two non-zero coefficients among the discriminative group of correlated variables[3], LAPS does exactly what we'd like, finding runs in both regions, and giving (lower, but) almost equal predictive weight to the whole group (which is found to be a single run).

### 2.2 K—"Soft" Fusion

LAPS models use non-identity $K$ matrices which provide them a very flexible set of modelling choices—all the way from strongly dependent run coefficients (via strong correlation structure of $K$) to models where the entire set of coefficients in the run is exchangeable ($K = \boldsymbol{I}$).

We parametrize these $K_j$ matrices by a single scalar $k$, based on the following assumptions. First, we assume that

[1] The Lasso uses $L_1$-regularization to achieve simultaneous sparsity and complexity control. Genkin et al. [2] and others report excellent predictive performance in high dimensional applications within this modelling framework.

[2] This matrix norm penalty is the Mahalanobis distance of the run coefficients to the (appropriate-size) zero vector. In a Bayesian interpretation, zero is the location of the prior mode and the prior covariance matrix involves, $K_j^{-1}$. Details can be found in Appendix A. Such matrix norms were suggested by Yuan and Lin [10] (and in their references), but all their subsequent modelling and experiments used only the identity matrix.

[3] Since the Lasso models all coefficients as exchangeable and strongly favors parsimony, this behavior is expected.

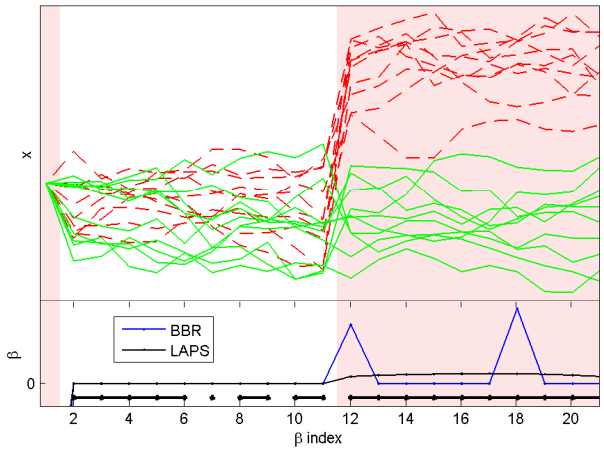


**Figure 2. Simple example showing proof of concept. In the top portion we plot the data set used (created by collating different 10-dimensional highly correlated Gaussians as shown. The correlation boundaries define groups, which are shaded. 20 samples total, 10 from each class). The two classes are shown in different colors. The bottom portion of the plot shows the Lasso model coefficients (found using BBR [2] with hyperparameter selected by 10-fold CV) and the estimated LAPS model coefficients and inferred run structure, $\mathcal{I}$ (denoted by the bands below the coefficients). The thin shaded region on the left is for the intercept term, and its coefficient is cropped out of the y-axis of the bottom portion.**

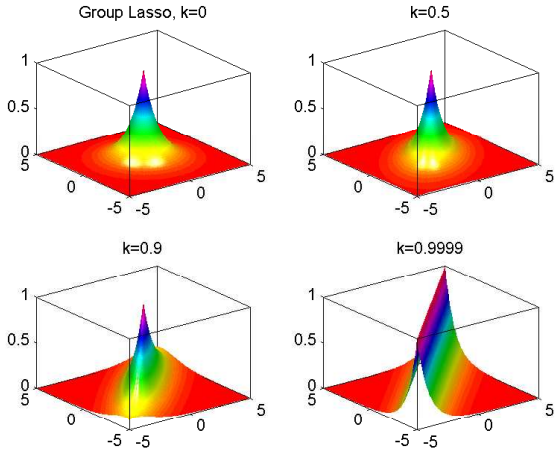


**Figure 3. The effect of $k$ on the prior for 2-D—illustrating soft fusion. Notice how as $k$ increases, prior mass shifts favoring both parameters to be more like each other.**

*a-priori* all the components of $\boldsymbol{\beta}$ have equal variance, regardless of the size of the run they will be in (a Bayesian interpretation is involved, see Appendix A). Second, since we seek runs on ordered variables, we try to impose the requirement that consecutive model coefficients in the same run should be similar. We accomplish this via tri-diagonal $K_j$'s. The corresponding $K_j^{-1}$ is a symmetric positive definite matrix with ones on the diagonal (this ensures the equal variance of components) and terms in decreasing geometric progression (multiplicative factor $k$) proportional to the distance from the diagonal (a Green's matrix).

See Figure 3 for a graphical view of how the Bayesian prior varies with respect to the fusion parameter $k$ on a two-variable size run. The $k$ value rather intuitively controls how much soft fusion we enforce (for $k$=0, we obtain the group Lasso). For a slightly bigger example, consider the matrices obtained for a run of size 4, with $k$=0.5. Here:

$$K^{-1} = \begin{pmatrix} 1 & 0.5 & 0.25 & 0.125 \\ 0.5 & 1 & 0.5 & 0.25 \\ 0.25 & 0.5 & 1 & 0.5 \\ 0.125 & 0.25 & 0.5 & 1 \end{pmatrix}, \text{and}$$

$$K = \begin{pmatrix} 1.333 & -0.667 & 0 & 0 \\ -0.667 & 1.667 & -0.667 & 0 \\ 0 & -0.667 & 1.667 & -0.667 \\ 0 & 0 & -0.667 & 1.333 \end{pmatrix}.$$

## 3 Learning LAPS models

We describe the algorithm for fitting full LAPS models (with parameters $\boldsymbol{\beta}$, $\mathcal{I}$, $k$, and $\lambda$) in stages. First, consider the situation where values for $k$, $\lambda$ and the run structure $\mathcal{I}$ are known. In this case, given the labelled dataset $D$, as well as $\lambda$, $\mathcal{I}$ and $k$, we want to find $\boldsymbol{\beta}$ such that:

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, g_{\lambda,\mathcal{I},k}(\boldsymbol{\beta}) = nll(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{J} s_j \|\boldsymbol{\beta}_{\mathcal{I}_j}\|_{K_j}. \quad (2)$$

We will refer to this as the core LAPS optimization problem. It is a convex optimization problem and we use a standard block coordinate descent algorithm to solve it (see Algorithm 1, [4]. We use simple off-the-shelf line search and Newton solvers)[4]. Convexity is crucial, and the algorithm results from repeated application of the optimality criteria

[4]Although this is probably not the most efficient algorithm for this problem, in our experiments it proved to be quite reasonable.

**Algorithm 1** Core LAPS optimization problem

**Input:** Training data $D$, initial $\boldsymbol{\beta}$.
**Result:** $\boldsymbol{\beta}$ that satisfies Equation 2.
**repeat**
  $\beta_0 \leftarrow \text{argmin}_{\beta_0}\, g_{\lambda,\mathcal{I},k}(\boldsymbol{\beta})$ (Line search for intercept).
  **for** $j = 1$ **to** $J$ **do**
    **if** $\left\|\nabla nll(\boldsymbol{\beta})_{\mathcal{I}_j}\right\|_{K_j^{-1}} \leq \lambda s_j$ **then**
      $\boldsymbol{\beta}_{\mathcal{I}_j} \leftarrow 0$
    **else**
      $\boldsymbol{\beta}_{\mathcal{I}_j} \leftarrow \text{argmin}_{\boldsymbol{\beta}_{\mathcal{I}_j}}\, g_{\lambda,\mathcal{I},k}(\boldsymbol{\beta}_{\mathcal{I}_j})$ (by Newton's method, say).
    **end if**
  **end for**
**until** Some convergence criteria is met.

(Appendix B provides a sketch of the derivation). Note that throughout we do not penalize the intercept term $\beta_0$. Next, consider the search for $\mathcal{I}$ (with $\lambda$ and $k$ still fixed).

### 3.1 $\mathcal{I}^*$—Run Structure Search

Motivated by the work of Consonni and Veronese [1], we use a heuristic greedy procedure for the run structure search. The search starts at an initial $\mathcal{I}$ derived using the core LAPS optimization problem (see Appendix D for details). The search then proceeds by locally perturbing an existing partition structure to generate a candidate run structure (see Figure 4). We then obtain the optimal $\boldsymbol{\beta}^*$ corresponding to this candidate. We use the optimal objective function value, $g_{\lambda,\mathcal{I},k}(\boldsymbol{\beta}^*)$ (Equation 2) to score the candidates. The search stops when all changes to the existing run structure result in worse scoring models. Note that our greedy search just involves repeated solutions of the (efficient) core LAPS optimization problem. Further, since the amount of regularization is fixed at this stage ($\lambda$, $k$ constant), the optimal objective function value, $g$, makes a sensible scoring criterion for the models.

### 3.2 Selecting the Hyperparameters

Finally, we propose to search over a discrete grid for the remaining hyperparameters, $\lambda$ and $k$. For each ($\lambda$,$k$) pair, we greedily search over the run structures for locally best $\mathcal{I}$ and $\boldsymbol{\beta}$ pairs (as outlined in subsection 3.1).

Having found $\boldsymbol{\beta}^*$ and $\mathcal{I}^*$ for every ($\lambda$,$k$) pair, we now score these locally "optimal" models. We cannot use the same $g$ function value for scoring *across* the different grid points as they have different amounts of regularization. A cross-validation accuracy based score for instance, makes sense, but may prove computationally infeasible for even moderate size problems (this would require repeated solutions of the $\mathcal{I}^*$ search problem for each fold). We instead use an approximate marginal data likelihood[5] based score (Appendix C).

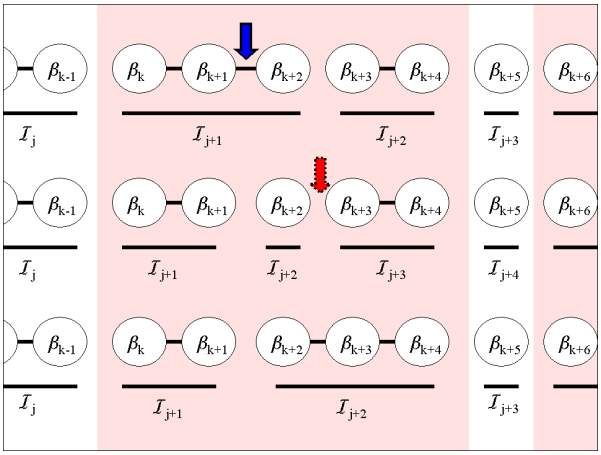


**Figure 4. Illustrating the $\mathcal{I}^*$ search with a graphical representation of the model coefficients. Edges (parts of a run) can only occur between adjacent nodes (coefficients) within the same group (shaded regions, as before). The run structure $\mathcal{I}$, is the set of all connected components defining the runs, $\mathcal{I}_j$s. Our search strategy works by sequentially examining all potential or existing edges. Two accepted perturbations and the corresponding run partitions are shown for the middle group (from the top to bottom rows).**

$$\begin{aligned} S(\boldsymbol{\beta}^*, \mathcal{I}^*, \lambda, k) \;=\; & nll(\boldsymbol{\beta}^*) + 0.5 \log|\Psi(\boldsymbol{\beta}^*, \mathcal{I}^*)| \\ & + \lambda \sum_{j=1}^{J} s_j \|\boldsymbol{\beta}^*_{\mathcal{I}^*_j}\|_{K_j} - \sum_{j=1}^{J^*} \log\left(\frac{c(|\mathcal{I}^*_j|)}{|\Sigma_j|^{0.5}}\right) \\ & - 0.5(nJ^* + 1)\log(2\pi), \end{aligned}$$

where $\Psi(\boldsymbol{\beta}, \mathcal{I}) = X^T A X + \lambda \sum_{j=1}^{J^*} s_j (\|\boldsymbol{\beta}_{\mathcal{I}_j}\|^{-1}_{K_j} K_j - \|\boldsymbol{\beta}_{\mathcal{I}_j}\|^{-3}_{K_j} B)$, with $A$ being the $t \times t$ diagonal matrix formed by the $a_{ii} = \Phi(\boldsymbol{\beta}^{*T}\mathbf{x}_i)(1 - \Phi(\boldsymbol{\beta}^{*T}\mathbf{x}_i))$ terms, $B = \boldsymbol{b}\boldsymbol{b}^T$ (an outer product of $\boldsymbol{b} = K_j \boldsymbol{\beta}^*_{\mathcal{I}^*_j}$ vectors. The summation till $J^*$ is only over the non-zero $\boldsymbol{\beta}^*$ run indices (and $nJ^*$ is the number of non-zero $\boldsymbol{\beta}^*$ indices). Appendix A has expressions for $c, \Sigma_j$.

We compare the model scores $S(\boldsymbol{\beta}^*, \mathcal{I}^*, \lambda, k)$ over the entire ($\lambda$,$k$) grid and pick the best model amongst those (as defined, smaller $S$ is better. This then results in values for

[5] Recall that the marginal data likelihood or evidence is proportional to the probability of the hypothesis (the particular hyperparameter settings in our case) given the data, and is a standard Bayesian model selection criterion.

$\lambda^*$ and $k^*$). In our experiments we have found this procedure quite robust to variations in the dataset and parameter settings.

## 4 Experiments

We next apply LAPS models to real and simulated data. We are interested in evaluating both structural (run partitioning) performance and predictive accuracy. Predictive performance comparisons are made to Lasso logistic regression (using BBR[6], [2], publicly available software).

### 4.1 SIM Data

In our first set of experiments we simulate datasets from three different models, with regression coefficients designed to favor one of regular Lasso, the group Lasso, and LAPS. The data $\mathbf{x} \sim N(0,1)^{15}$, are simulated to be uncorrelated 15-dimensional Gaussian with mean zero and unit variance. The $\boldsymbol{\beta}_{true}$ are shown in Table 1. A large test dataset ($10^4$ examples) was also simulated for each set of regression coefficients. As can be seen, each set of coefficients favors one of the three models—the first set has no intentionally long runs (thus favors the Lasso model, SIM1), the second set has runs, but with no internal similarity (thus favors the group Lasso, SIM2), and the third has runs with extremely high similarity (SIM3). In all three cases, the coefficients are scaled such that the Bayes risk, $r = \sum_{i=1}^{t} \min\{\Phi(\boldsymbol{\beta}_{true}^T \mathbf{x}_i), 1 - \Phi(\boldsymbol{\beta}_{true}^T \mathbf{x}_i)\}$, on the large corresponding test dataset is 0.2. In order to assess sample size effects, we also simulate training datasets of two sizes, small (50 examples, denoted in the results by SM) and large (LG, 500 examples). There is only one group here, which corresponds to the entire set of predictors, [1—16] (Index 1 is for the intercept).

The results show that LAPS does a fairly reasonable job of finding non-zero coefficient runs[7] even with the small datasets, and performs much better on the large datasets (see the $\mathcal{I}$ bands in the plots in Figure 5). The inferred $k^*$ parameter also provides insight, being 0 and 0.99 for SIM2-LG and SIM3-LG, indicating run structure that is not-at-all and highly similar respectively. Also, predictively, LAPS performs competitively with the Lasso (see Table 2), having small gains and losses in the expected cases.

[6] `http://www.stat.rutgers.edu/~madigan/BBR`

[7] We point out here that runs consisting only of zero coefficients aren't necessarily grouped correctly because the the $\mathbf{x}$s are uncorrelated and the score we use $g$, is insensitive to zero-coefficients being grouped in a run together (MAP zero coefficients in any number of runs result in the same $g$ value).

**Table 1. SIM data regression coefficients ($\boldsymbol{\beta}_{true}$, in columns). The first index corresponds is for the intercept term. The desired runs are shown as blocks in the columns (same as the blue bands at the bottom of the relevant plots in Figure 5).**

| $\boldsymbol{\beta}$ Index | SIM1 | SIM2 | SIM3 |
|---|---|---|---|
| 1 | 0 | 0 | 0 |
| 2 | 1.1500 | 0 | 0 |
| 3 | 0 | -1.1609 | -0.9540 |
| 4 | 0.5750 | 0.5804 | -0.9540 |
| 5 | -0.2875 | -0.8706 | -0.9540 |
| 6 | 0 | 0.5804 | -0.9540 |
| 7 | 0 | 0 | 0 |
| 8 | -0.2875 | 0 | 0 |
| 9 | 0.5750 | 0 | 0 |
| 10 | 0 | -0.5804 | -0.4770 |
| 11 | 1.1500 | 0.2902 | -0.4770 |
| 12 | 0 | -1.1609 | -0.4770 |
| 13 | -1.1500 | 0 | 0 |
| 14 | 0 | 0 | 0 |
| 15 | 0 | 0.8706 | 0.7155 |
| 16 | -0.8625 | -0.2902 | 0.7155 |

### 4.2 BF Data

The cylinder, bell, funnel dataset proposed by Saito [6] is a three class problem, with one input group variable per example. The equations describing the $\mathbf{x}$ for each class have both random noise as well as random start and end points for the generating events of each class, making for quite a lot of variability in the instances. We focus on just two classes, bell vs. funnel. The class labels roughly describe the shape of the examples—bells ramp up and then drop at some point sharply, and funnels spike sharply and then ramp down. As in past studies, we simulated 266 instances of each class to construct the dataset. The top two rows of Figure 6 show the data and a particular instance of each class in bold (bells in red and funnels in blue). Once again, there is only one big group here, consisting the entire set of variable indices (128 predictors).

Both LAPS and Lasso make just one error on one fold in 10-fold cross-validation predictive accuracy experiments (Table 2). Indeed the best LAPS model is very similar to the best Lasso model, with $k^*$=0, see Figure 6. However, some salient properties of the dataset become apparent when examining successive LAPS models with increasing run cohesiveness (i.e., fixing at $\lambda^*$, increasing $k$). As can be seen in the figure, the $k = 0.99$ model seems to imply three specific

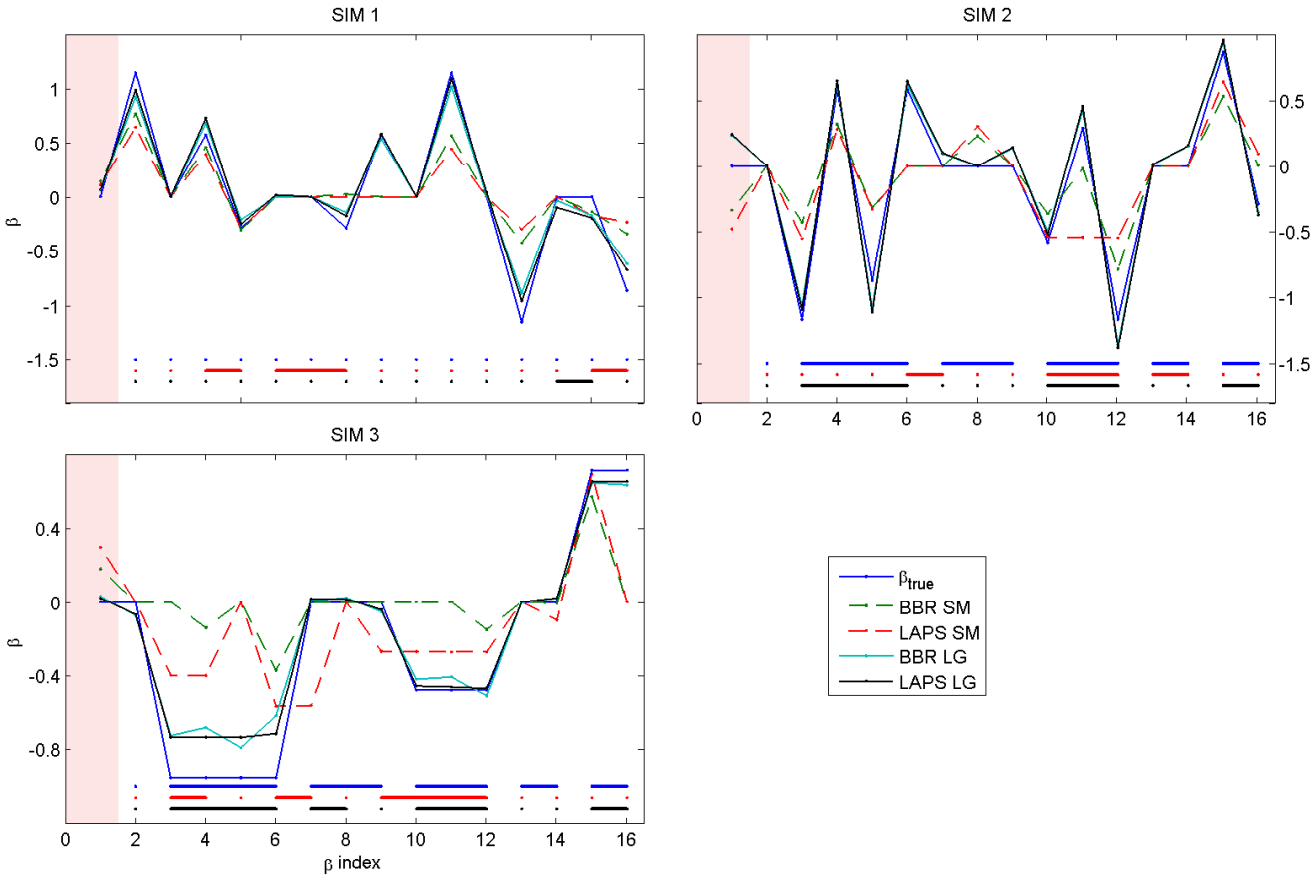


**Figure 5. Results on the SIM datasets. The SIM plots show the true, Lasso and best LAPS $\beta$s and LAPS $\mathcal{I}$ (the bands at the bottom of each plot). The intended/true run structure for the SIM data is also shown in blue bands.**

runs which are discriminative—the first run is not so strong, occurs early (indices around 15—30) and primarily focuses on "early" rising bells from the funnels. The next run is the most significant, occurs right afterwards (from about 35—40) and is the region of the data where the two classes are most segregated[8]. Finally, a short positive run around 50 works in combination with the previous two runs—by this index location, a bell should be on the ascendency compared to a funnel.

### 4.3 NHP study

Our final application concerns a vaccine efficacy study. 136 non-human primates (NHPs) were vaccinated, monitored for a year and then "challenged" with anthrax. Of the 136 NHPs, 93 survived the challenge and 43 died. Repeated measurements during the year (and some up to a year after) assessed over a dozen aspects of the putative immune response. These measurements include an include an immunoglobulin G enzyme-linked immunosorbent assay (IgG), various interleukin measures (IL2, IL4, IL6), and a so-called "stimulation index" (SI), to name a few, with the number of measurements varying somewhat from animal to animal. The goal of the study is to understand the predictive value of the various assays with respect to survival. The assays thus define the groups, and we search for runs in their time series measurements.

The best LAPS model found with $k^* = 0$, looks a lot like the best Lasso model found. Again, it is instructive to look at the LAPS models obtained by varying $k$ (Figure 7, holding $\lambda^*$ fixed). The models are biologically reasonable—high amounts of antibodies that neutralize the toxin predict survival (IgG, ED50/TNA). High amounts of interleukins (immune system response/signaling molecules), particularly as time increases, are predictive of death (il4eli, il2m,

[8]While the region around 80—100 also appears to be similarly segregated, a careful look reveals it is actually much more mixed, because some examples for both classes have "fallen" trajectories by this index location.

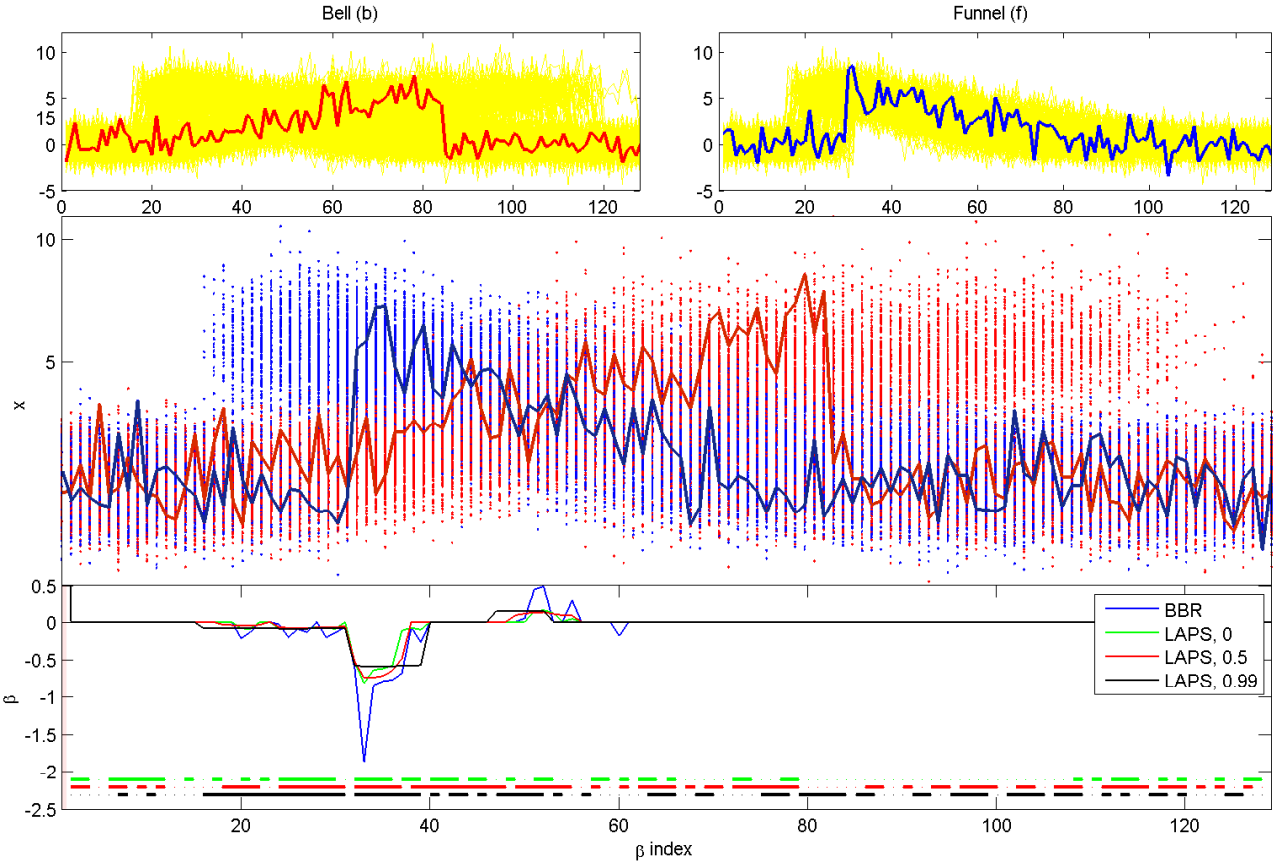


**Figure 6. Results on the bell-funnel data. The plots in the top most row show all bell and funnel instances in yellow along with one instance each highlighted. The plot in the second row also shows the whole dataset (and one highlighted bell and funnel instance) but with no lines connecting the data points for clarity. The bottom portion plots the Lasso and best LAPS $\beta$ and corresponding $\mathcal{I}$s. For the other LAPS models in this plot, $\lambda$ is held fixed and $k$ is varied.**

il4m etc.). Finally, the model also allows one to find runs that are likely not predictive—see for example the first half of the il6m assay, which is identified as a single run across different the $k$ values, and consistently set to zero.

## 5 Discussion

In this paper we present a model based on the Lasso for finding predictive runs in particular types of structured classification problems. We provide the details of the model, an algorithm for inferring its parameters, and results of application on different types of data. Many extensions of the current work are possible, of which we mention a few. Viewed in Bayesian framework, in this work we have assumed a flat prior over partition space—it would be interesting to see the effect of various priors on run-partition

**Table 2. Predictive performance—estimated error rates**[10]

| Data | Lasso | | LAPS | | |
|---|---|---|---|---|---|
| | % Err | $V^*$ | % Err | $V^*$ | $k^*$ |
| SM1 | **25.43** | 0.45 | 27.52 | 0.28 | 0 |
| SM2 | **30.83** | 0.15 | 34.38 | 0.54 | 0.99 |
| SM3 | 35.98 | 0.15 | **30.62** | 0.37 | 0.99 |
| LG1 | 22.31 | 0.15 | **22.09** | 0.54 | 0.74 |
| LG2 | 21.14 | 0.5 | **21.09** | 0.63 | 0 |
| LG3 | 21.86 | 0.35 | **21.68** | 0.19 | 0.99 |
| BF | $0.1887 \pm 0.6$ | 200 | $0.1887 \pm 0.6$ | 0.45 | 0 |
| NHP | $30.81 \pm 11.97$ | 0.2 | **28.02** $\pm 10.27$ | 0.46 | 0 |

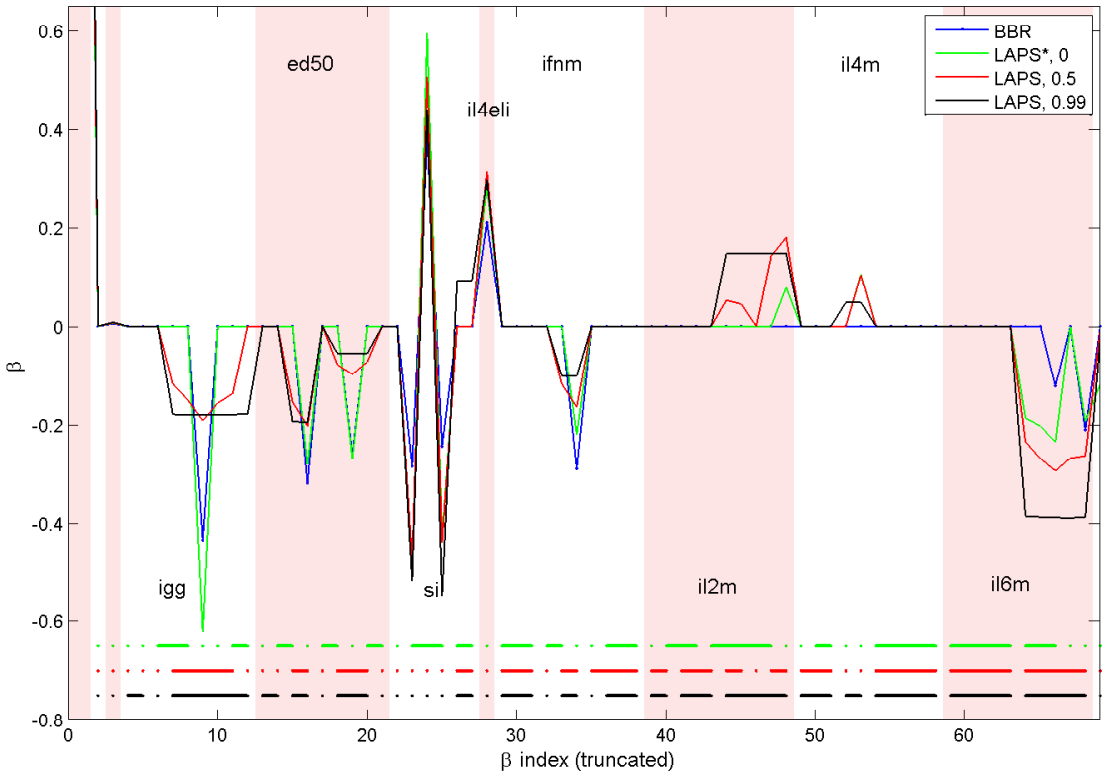


**Figure 7. Lasso vs. LAPS on the NHP dataset.**

space. In particular, a non-parametric prior like the Chinese restaurant process prior (adapted for the ordering of variables), might provide for an alternative way for the data to "decide" on the number and type of runs. One could also look at alternatives to the MAP style Bayesian analysis carried out here—numerical simulation may be used to generate a posterior distribution on LAPS model outputs. Finally, extending this model to larger problems (in the number of attributes, the number of training examples and also dimension of structured data—2-D images, for example) raises interesting computational and methodological issues.

## A: The LAPS prior

Viewed as a Bayesian prior, the LAPS regularization term corresponds to a product of multivariate power exponential (MVPE) distributions with power 0.5 [3]. The particular MVPE distribution we use has the following density function (for a particular run and with mean zero):

$$f(\boldsymbol{\beta}_{\mathcal{I}_j}|\mu=\mathbf{z},\Sigma_j)=\frac{c(|\mathcal{I}_j|)}{|\Sigma_j|^{1/2}}\exp-\frac{1}{2}\left[\boldsymbol{\beta}_{\mathcal{I}_j}^T\Sigma_j^{-1}\boldsymbol{\beta}_{\mathcal{I}_j}\right]^{0.5},$$

where the normalizing constant contains $c(|\mathcal{I}_j|) = \frac{|\mathcal{I}_j|\Gamma(|\mathcal{I}_j|/2)}{\pi^{|\mathcal{I}_j|/2}\Gamma(1+|\mathcal{I}_j|)2^{1+|\mathcal{I}_j|}}$. The covariance matrix of this distribution is given by: $\mathrm{cov}(\boldsymbol{\beta}_{\mathcal{I}_j}) = \frac{4\Gamma(nj+2)}{nj\Gamma(nj)}\Sigma_j = 4(n+1)\Sigma_j,\ \forall|\mathcal{I}_j|\in\mathbb{Z}$ ($|\mathcal{I}_j|$ denotes the size/cardinality of $\mathcal{I}_j$). For LAPS, we set $\Sigma_j^{-1} = 2(nj+1)\lambda^2K_j$. This results in $\mathrm{cov}(\boldsymbol{\beta}_{\mathcal{I}_j}) = 2K_j^{-1}/\lambda^2$. Since the $K_j$ matrices have unit diagonals, the marginal prior variance of every parameter is identical (and equal to $2/\lambda^2$, which is exactly the Lasso model prior variance). This $\Sigma_j$ setting then results in $s_j=\sqrt{0.5(|\mathcal{I}_j|+1)}$.

Also, the $K_j$ being tri-diagonal results in approximate structural conditional independence[11]. This can be seen by examining an approximating Gaussian graphical model, which would be found by matching the first and second moments (exactly as above). The structural zeros in the

[10]The error estimates are obtained from the test set for the SIM data (SM/LG 1. . .3) and by 10-fold CV on BF and NHP( at $k^*, V^*$ with standard error shown). $V = 2/\lambda^2$, is an equivalent paramterization of $\lambda$. $V^*$ (and $k^*$, for LAPS) are found through grid search. The grid for $k$ is consistently set to five values uniform over the range [0—0.99] (including both end points). The search ranges for $V$ for Lasso and LAPS are [0.01—1] and [0.1—0.9] respectively for the SIM datasets . For the BF dataset, the ranges were [0.01—$10^3$] and [0.01—1], and [0.05—1], [0.1—0.9] for the NHP data (Lasso and then LAPS respectively). The Lasso $V$ grid was always chosen at least thrice as fine as the uniform 10 grid points from the interval for LAPS.

[11]True conditional independence is not possible for any non-diagonal inverse covariance due to the 0.5 power in the exponent.

approximating Gaussian's inverse covariance result in runs being a linear chain graphical models.

## B: Core LAPS problem optimality criteria

There are two distinct optimality cases to check run-wise for a purported solution $\boldsymbol{\beta}^*$ of Equation 2 ($\mathcal{I}$ is given): One, if the run is set to zero $\beta^*_{\mathcal{I}_j} = 0$, and two, if all the elements in the run are non-zero $\beta^*_{\mathcal{I}_j} \neq 0$. If $\beta^*_{\mathcal{I}_j} \neq 0$ for the run, the optimality conditions are derived by simply setting the gradient of the objective function to zero.

If $\beta^*_{\mathcal{I}_j} = 0$, we need convex non-smooth analysis results [5] because the regularization term is non-differentiable at zero. We will use both the notions of the subgradient (a tangent plane supporting a convex function, $f$. Precisely, the subgradient $\xi \in \mathbb{R}^{|z|}$, of $f$ at $z_0$ is defined to be any vector satisfying: $f(z) \geq f(z_0) + \xi^T(z - z_0)$) and the subdifferential ($\partial f$ which is the set of all subgradients, $\xi$, at a point. This collapses to the ordinary derivative when the function is differentiable.). We will also use the following theorem: *$\hat{\beta}$ is a global minimizer of a convex function $f(\beta)$ if and only if* $0 \in \partial f(\hat{\beta})$.

Now for $\beta^*_{\mathcal{I}_j} = 0$, use the theorem above. For the matrix norm part of the objective function, the subgradient $\xi \in \mathbb{R}^{|\mathcal{I}_j|}$ satisfies $\left\|\beta_{\mathcal{I}_j}\right\|_{K_j} \geq \xi^T \beta_{\mathcal{I}_j}$ (by the definition). Now, consider the (unique) Cholesky decomposition of the positive definite matrix $K_j = R_j^T R_j$ (also define $\alpha_j = R_j \beta_{\mathcal{I}_j}$). Rewriting the previous condition, we can express the subdifferential as the set $\xi_j$: $(\beta^T_{\mathcal{I}_j} K_j \beta_{\mathcal{I}_j})^{0.5} = (\alpha_j^T \alpha_j)^{0.5} = \|\alpha_j\|_2 \geq \xi_j^T \beta_{\mathcal{I}_j} = \xi^T R_j^{-1} \alpha_j$. This inequality in turn, holds whenever $\left\|(R_j^{-1})^T \xi_j\right\|_2 \leq 1$, which can also be seen to be equivalent to $\|\xi_j\|_{K_j^{-1}} \leq 1$ (because $K_j^{-1} = (R_j^T R_j)^{-1} = R_j^{-1}(R_j^T)^{-1}$). The theorem then requires that $0 \in \nabla nll(\boldsymbol{\beta})_{\mathcal{I}_j} + \lambda s_j \{\xi_j : \|\xi_j\|_{K_j^{-1}} \leq 1\}$ be satisfied. This finally yields: $\left\|\nabla nll(\boldsymbol{\beta})_{\mathcal{I}_j}\right\|_{K_j^{-1}} \leq \lambda s_j \quad \forall \boldsymbol{\beta}_{\mathcal{I}_j} = 0$. Thus the optimality criteria result:

$$\nabla nll(\boldsymbol{\beta})_{\mathcal{I}_j} + \lambda s_j \frac{K_j \boldsymbol{\beta}_{\mathcal{I}_j}}{\|\boldsymbol{\beta}_{\mathcal{I}_j}\|_{K_j}} = 0 \quad \forall \boldsymbol{\beta}_{\mathcal{I}_j} \neq 0,$$
$$\left\|\nabla nll(\boldsymbol{\beta})_{\mathcal{I}_j}\right\|_{K_j^{-1}} \leq \lambda s_j \quad \forall \boldsymbol{\beta}_{\mathcal{I}_j} = 0. \qquad (3)$$

## C: The Approximate Marginal Data Likelihood Score

The $S$ score we use, is based on a Laplace approximation to the posterior distribution. As anticipated, the non-differentiability of the regularization term at zero complicates evaluating it. We use an approximation suggested in Shimamura et al. [7], which essentially ignores the contribution to the curvature of the posterior by the $\beta^*_{\mathcal{I}_j^*} = 0$ components (in other words, performs a Laplace approximation by only considering the non-zero coefficients/variables—we denote these by the summation till $J^*$, instead of $J$ for all the attributes). This rather drastic appearing assumption and a straightforward second order Taylor expansion of the negative log posterior results in the score we use. We note here that the assumption is really not as bad as it appears on first glance. Indeed, the posterior is clearly less curved along zero coefficient axes—this can be seen from the optimality conditions, Equation 3, by looking at the magnitude of the subdifferential in both cases. Further, simulation studies also show this approximation to be quite reasonable in practice.

## D: Heuristic for the initial $\mathcal{I}$

Given $k$ and $\lambda$ the procedure (for a single group) is: 1. For each attribute, fit a group Lasso model to the outputs with the current neighbors as the only predictors (a list initially containing just the attribute itself). If the minimum $g_{\lambda,\mathcal{I},k}$ value (restricted to only the attributes in the list) of an expanded neighborhood is better than that for the old neighborhood, update the neighborhood. Otherwise if all expansions result in poorer $g$ value, stop and record the final neighborhood for that variable. 2. Once the neighborhoods for all the variables (in the group) have been obtained, the initial run partition is given by the set of maximal cliques (the largest subgroups where each variable in the clique votes to have the other variable in it's neighborhood, and vice-versa). In our experiments, this heuristic performs quite well at very reasonable computational cost.


### Acknowledgements

We thank the US National Science Foundation for financial support through grants DMS-0505599 and EIA-0087022.


## References


[1] G. Consonni and P. Veronese. A bayesian method for combining results from several binomial experiments. *Journal of the American Statistical Association*, 90: 935 – 944, 1995.

[2] A. Genkin, D. D. Lewis, and D. Madigan. Large-scale bayesian logisitic regression for text categorization., 2004. URL `http://www.stat.rutgers.edu/ madigan/PAPERS/`.

[3] J. K. Lindsey. Multivariate elliptically contoured distributions for repeated measurements. *Biometrics*, 55 (4):1277 – 1280, 1999.

[4] L. Meier, van de Geer, S, Bühlmann, and P. The group lasso for logistic regression. Technical report, ETH, Zurich, Switzerland, 2006.

[5] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, N.J, 1970.

[6] N. Saito. *Local feature extraction and its application using a library of bases.* PhD thesis, Yale University, 1994.

[7] T. Shimamura, H. Minami, and M. Mizuta. Regularization parameter selection in the group lasso. In *COMPSTAT*, Capri, Italy, 2006.

[8] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society*, 67(1): 91 – 108, 2005.

[9] R. J. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1):267–288, 1996.

[10] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society*, 68(Series B):49 – 67, 2006.