The Partitioned LASSO-Patternsearch Algorithm with Application to Gene Expression Data

Weiliang Shi, Grace Wahba, Rafael Irizarry,
Hector Corrada Bravo, Stephen Wright

August 23, 2011

Abstract

The Partitioned LASSO-Patternsearch algorithm is proposed to identify patterns of multiple dichotomous risk factors for outcomes of interest in genomic studies. A partitioning scheme is used to identify promising patterns by solving many LASSO-Patternsearch subproblems in parallel. All variables that survive this stage proceed to an aggregation stage where the most significant patterns are identified by solving a reduced LASSO-Patternsearch problem in just these variables. This approach was applied to genetic data sets with expression levels dichotomized by gene expression bar code. Most of the genes and second-order interactions thus selected and are known to be related to the outcomes. Cross-validation shows that the proposed method provides smaller models with better prediction accuracy, in comparison to several competing methodologies.

1 Introduction

The LASSO-Patternsearch (LPS) algorithm [19, 12, 23] is an effective approach for identifying multiple dichotomous risk factors for outcomes of interest in demographic and genomic studies. It uses an ℓ_1-regularized logistic regression formulation, targeting the case in which only a small fraction of the large number of possible candidate patterns are significant. The

approach can be used to consider simultaneously all possible patterns up to a specified order. It can identify complicated correlation structures among the predictor variables, on a scale that can cause serious difficulties for algorithms that target problems of more modest size.

When applied to very large models with higher-order interactions between the predictor variables, however, LPS quickly runs into computational limitations. For example, a problem with two thousand predictor variables yields a logistic-regression formulation with about two million variables if both first- and second-order patterns are included in the model. Problems of this size are at the limit of LPS capabilities, yet current problems of interest in genetic epidemiology consider ten thousand markers or more [21]. For these kinds of data sets, a screening stage can be added before applying LPS [18].

In this article, we propose a Partitioned LASSO-Patternsearch Algorithm (pLPS) scheme to tackle gigantic data sets in which we wish to consider second- and possibly third-order interactions among the predictors, in addition to the first-order effects. As in LPS, we assume that all predictor variables are binary (or that they have been dichotomized before the analysis). The model thus contains a huge number of possible patterns, but the solution is believed to be sparse, with only a few effects being significant risk factors for the given outcome. In the first (screening) stage of pLPS, the predictors are divided into partitions of approximately equal size, and LPS is used to solve smaller subproblems in which just the predictors and higher-order effects within a single partition, or the interactions between variables in small groups of partitions, are considered as variables in the optimization model. These reduced problems can be solved independently, in parallel. By the end of the screening stage, each predictor and each higher-order effect (up to the specified order) has been considered in at least one of the subproblems. The second stage of pLPS is an aggregation process, in which all predictors identified in the first stage are considered, together with all their interactions up to the specified order. An LPS process is used to identify the final set of significant predictors and interactions.

Tuning parameters in the first stage of pLPS are chosen by BGACV criterion (see [19]). In the second stage, two tuning parameters are used, one for main effects and one for interactions. These are chosen by BGACV2, a variation of BGACV to be described below. We examine the effectiveness of the pLPS strategy on simulated data and on two large-scale genetic data sets.

The rest of the article is organized as follows. In Section 2 we describe the details

of the pLPS algorithm. Section 3 presents three simulation examples that demonstrate the properties of pLPS and contrast its results with those obtained from Logic Regression, SPLR, and Random Forest. Section 4 applies the method to gene expression data, dichotomized by the bar code method [24]. Section 5 presents some conclusions.

2 Partitioned LASSO-Patternsearch Algorithm

We now give further details of the pLPS scheme and its implementation. For simplicity, most of our discussion focuses on the case in which first-order effects and second-order interactions between all predictors are considered. Extension of the approach to include third-order effects as well is described briefly at the end of the section.

Considering n subjects with p binary predictor variables, the total number of interactions up to order q is given by $N_B = \sum_{\nu=0}^{q} \binom{p}{\nu}$. For $q = 2$, we thus have $1 + p(p+1)/2$ patterns. To apply pLPS, we first divide the p variables into k partitions so that each partition has $g = p/k$ variables. (For simplicity of description, we assume that p is divisible by k.) The data set is $\{y, x_j, j = 1, 2, \ldots, p\}$, where $y = (y_1, y_2, \ldots, y_n) \in \{0, 1\}$ is the response, $x_j = (x_j(1), x_j(2), \ldots, x_j(n))$ is the jth covariate, and $x_j(i) \in \{0, 1\}$ for all $j = 1, 2, \ldots, p$ and $i = 1, 2, \ldots, n$. By relabelling the p predictors as x_{st}, where $s = 1, 2, \ldots, k$ denotes the partition number and $t = 1, 2, \ldots, g$ denotes the index within the partition, we relabel the full data set as $\{y, x_{st},\ s = 1, 2, \ldots, k,\ t = 1, 2, \ldots, g\}$.

In the first stage of pLPS (the "screening stage"), we solve two types of reduced LPS subproblems. The first type is based on a pair of partitions, denoted by s_1 and s_2, and defines the LPS variables in the subproblems to be the first-order effects within each group (for which there are $2g$ basis functions $\{B_{t_1} = x_{s_1 t},\ t = 1, 2, \ldots, g\}$ and $\{B_{t_2} = x_{s_2 t},\ t = 1, 2, \ldots, g\}$) and all the second-order interactions between a predictor in group s_1 and a predictor in group s_2. There are g^2 basis functions for the latter effects, namely, $\{B_{t_1 t_2} = x_{s_1 t_1} \times x_{s_2 t_2},\ t_1, t_2 = 1, 2, \ldots, g\}$. Hence the total number of patterns in the LPS model for each subproblem is $g^2 + 2g + 1$, when we include the constant basis function $B \equiv 1$.

The second type of reduced LPS problem is obtained from the first- and second-order effects within a single partition. Here, the basis functions for group s are $\{B_{t_1 t_2} = x_{st_1} \times x_{st_2},\ t_1, t_2 = 1, 2, \ldots, g\}$ and $\{B_t = x_{st},\ t = 1, 2, \ldots, g\}$, making a total of $1 + g(g+1)/2$ patterns, when we include the constant basis function. Since each subproblem of the

second type has about half as many variables as each subproblem of the first type, we define computational tasks of roughly equivalent complexity by grouping two of the type-two problems together. Figure 1 is a graphical presentation of the two types of groups considered in the first stage of pLPS.

We now briefly describe the LPS methodology, which is applied to each of these subproblems. By relabelling, we define the basis functions to be $B_\ell(x)$, $\ell = 1, 2, \ldots, N_B$. Defining $p(x) := \text{Prob}[y = 1|x]$ and the logit (log odds ratio) $f(x) := \log[p(x)/(1-p(x))]$, we estimate f by minimizing

$$I_\lambda(y, f) = \mathcal{L}(y, f) + \lambda J(f), \tag{1}$$

where $\mathcal{L}(y, f)$ is the negative log likelihood divided by n:

$$\mathcal{L}(y, f) = \frac{1}{n}\sum_{i=1}^{n}[-y_i f(x(i)) + \log(1 + e^{f(x(i))})], \tag{2}$$

with f being expressed as a linear combination of the basis functions

$$f(x) = \mu + \sum_{\ell=1}^{N_B-1} c_\ell B_\ell(x), \tag{3}$$

and the penalty function being defined by

$$J(f) = \sum_{\ell=1}^{N_B-1} |c_\ell|. \tag{4}$$

(We assume that the last basis function is the constant function 1, whose coefficient μ does not appear in J and is therefore not penalized.) The penalty parameter λ in (1) is chosen by BGACV. We then build a parametric logistic regression model on the remaining basis functions by minimizing(2) and selecting the best model via backward elimination with the BGACV criteria. More details are given in [19, Section 3].

If the outcomes can be predicted well using a small number of patterns, the number of patterns surviving the first stage of pLPS should be small. Suppose there are a total of p^* unique predictor variables in all these patterns. The second stage of pLPS — the "aggregation stage" — is an LPS problem in which just these predictors and all their second-order effects are the patterns. There will be $N_{B_1} = p^*$ basis function (denoted by $B_{1\ell}$) for the main effects and N_{B_2} $(= \binom{p^*}{2})$ basis functions (denoted by $B_{2\ell}$) for the second-order

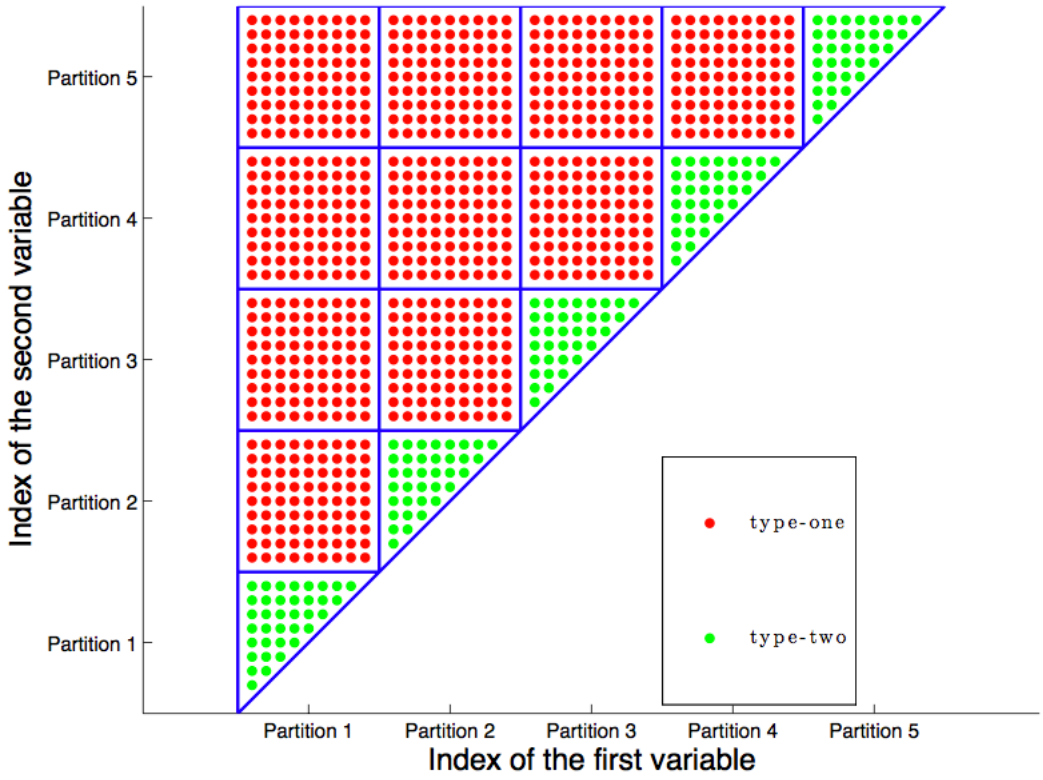

Figure 1: Diagram of the subproblems in the first stage of pLPS, assuming 5 partitions. Side length of a square is the partition size, while the horizontal axis contains the labels of the first effect and the vertical axis the label of the second effect. Squares filled with red dots are "type-one" subproblems while the triangles filled with green dots are "type-two" subproblems.

interactions, plus one constant basis function. In the aggregation stage, we use different penalty parameters for the first- and second-order patterns, so the objective function is

$$I_{\lambda_1,\lambda_2}(y, f) = \mathcal{L}(y, f) + \lambda_1 J_1(f) + \lambda_2 J_2(f), \tag{5}$$

where the link function f is

$$f(x) = \mu + \sum_{\ell=1}^{N_{B_1}} c_{1\ell} B_{1\ell}(x) + \sum_{\ell=1}^{N_{B_2}} c_{2\ell} B_{2\ell}(x), \tag{6}$$

and the penalties are

$$J_1(f) = \sum_{\ell=1}^{N_{B_1}} |c_{1\ell}|, \quad J_2(f) = \sum_{\ell=1}^{N_{B_2}} |c_{2\ell}|. \tag{7}$$

The choice of penalty parameters (λ_1, λ_2) in (5) is critical to the performance of this formulation. BGACV does not work well in this setting. Often, it tends to select only second-order patterns, combining main effects with spurious partners. Occasionally, it selects only main effects, breaking true size-two patterns into separate main effects. The large difference between the number of basis functions makes the solutions sensitive to the two penalty parameters. Searching over a grid of values for λ_1 and λ_2 is expensive and often does not give satisfactory results. As an alternative approach, we introduce the following penalty function, known as BGACV2:

$$BGACV2(\lambda_1, \lambda_2) = BGACV(\lambda_1, \lambda_2) \times \left(1 + 0.5 \frac{|n_{b_1} - n_{b_2}|}{n_{b_1} + n_{b_2}}\right), \tag{8}$$

where n_{b_1} is the number of nonzero coefficients of main effects and n_{b_2} is the number of nonzero coefficients of size-two patterns. The additional penalty factor forces these two numbers to be similar, reducing the possibility of the two extreme cases discussed above. If the true model only contains main effects, the BGACV2 penalty will tend to select fewer main effects than the BGACV model. However, BGACV is conservative (see discussion in [19]), while BGACV2 is less so. We expect that BGACV2 will not miss any important main effects, though it may also produce some spurious second-order effects. These spurious effects will be further eliminated by the parametric logistic regression step as noted above, followed by solving (5).

Minor extensions to the pLPS approach are needed when size-three patterns ($q = 3$) are introduced. In the screening phase of pLPS, there are four types of subproblems (rather than

two). These types are distinguished by considering the labels s_1, s_2, s_3 of the three partitions chosen to define the subproblem (with $s_1 \leq s_2 \leq s_3$). The four types correspond to the cases $s_1 < s_2 < s_3$, $s_1 = s_2 < s_3$, $s_1 < s_2 = s_3$, and $s_1 = s_2 = s_3$, respectively. In the aggregation phase of pLPS, we will still be using two penalty parameters, one for main effects and one for interactions; size-two and size-three patterns share the same penalty parameter. The criterion function for choosing the appropriate values for penalty parameters λ_1 and λ_2 is

$$BGACV3(\lambda_1, \lambda_2) = BGACV(\lambda_1, \lambda_2) \times \left(1 + 0.5 \frac{|n_{b_1} - n_a| + |n_{b_2} - n_a| + |n_{b_3} - n_a|}{n_{b_1} + n_{b_2} + n_{b_3}}\right), \quad (9)$$

where n_{b_1} is the number of nonzero main effects, n_{b_2} is the number of nonzero size two patterns, n_{b_3} is the number of nonzero size three patterns and n_a is the average of the three.

In the remainder of the paper, we use pLPS to denote the $q = 2$ case and pLPS3 for the $q = 3$ case.

The choice of g (the number of variables in each partition) is determined by the computing power and the available memory. On our super server (an AMD Dual-Core 2.8 GHz machine with 64 GB memory), we usually set $g = 2,000$ for $q = 2$. This yields subproblems with $N_B = 2,001,001$ basis functions, which can be handled comfortably by the LPS code. On a more standard computer (Intel(R) Pentium(R) 4 2.80GHz with 2 GB memory), we usually set $g = 200$ for $q = 2$ and $g = 35$ for $q = 3$. As we noted earlier, the subproblems in the first stage of pLPS can be solved independently in parallel, on different computers. The grid-computing system Condor (`http://www.cs.wisc.edu/condor/`) provides an ideal platform for these parallel jobs. In our Condor implementation, we request machines from the pool with at least 2 GB of memory, and define our group sizes to be $g = 200$ (for $q = 2$) and $g = 35$ (for $q = 3$). Generally, for faster execution of pLPS, it is advantageous to set g to the highest value that can be accommodated by the memory of the computer. The final results of the computation do not depend strongly on the choice of g.

3 Simulation Studies

In this section we study the empirical performance of pLPS through three simulated examples. The first example is a relatively small data set with independent predictor variables: One main effect and two second order interactions are included in the link function. The second example is a very large data set with strong correlations among neighboring variables, in

which two main effects and two second order interactions are assumed to be important. The third example studies the performance of pLPS3, which includes third-order interactions in the model. Two main effects, one second order interaction and one third order interaction are included.

We compare pLPS with three other methods:

- Logic Regression [17], as implemented in the R package `LogicReg`,
- Stepwise Penalized Logistic Regression (SPLR) [15], as implemented in the R package `stepPlr`, and
- Random Forest (RF) [1], as implemented in the R package `randomForest`.

The number of trees and number of leaves in Logic Regression are selected by five-fold cross validation. The smoothing parameter in SPLR is also selected by five-fold cross validation, and the model size selected by BIC.

3.1 Simulation Example 3.1

In our first example (Example 3.1), 400 iid Bernoulli(0.5) random variables were simulated. The sample size is 700 and the logit function is

$$f(x) = -2 + 1.5X_{50} + 1.5X_{150}X_{250} + 1.5X_{251}X_{252}.$$

One hundred data sets were generated according to this model and analyzed by the four methods described above.

Table 1 presents the results of this simulation. Each entry in the table shows the number of appearances of the pattern and the variables in the 100 simulations. The main number (outside the parentheses) is the *pattern count* showing how many times the given pattern is selected in the model. The numbers inside the parentheses are the *variable counts* showing how many times each variable in a given pattern appears in the model, either as a main effect or in some other interaction. Random Forest does not generate an explicit model, but rather produces an importance score for all variables. It is not possible to calculate a pattern count, but we calculate the variable count according to whether the variables in question appeared among the top 10 variables identified by Random Forest. For pLPS, Logic regression and SPLR, the last column labelled "noise" counts the total number of appearances in the 100

Table 1: Simulation Example 3.1 with $n = 700$ and $p = 400$, and no correlations. Tabulated numbers show the number of tests (out of 100) in which the pattern was detected by each algorithm. The number outside the parentheses is the number of times the given pattern was selected; the numbers inside the parentheses shows how many times the variables in the pattern are detected in the model, as a main effect or in some interaction. The final column shows the total number of times (in 100 tests) that the algorithms selected patterns (variables for RF) that do not appear in our model.

Methods	X_{50}	$X_{150}X_{250}$	$X_{251}X_{252}$	noise
pLPS	94 (100)	99 (99,99)	96 (97,97)	153
Logic	100 (100)	70 (88,91)	65 (84,90)	190
RF	NA (100)	NA (96,97)	NA (94,96)	(517)
SPLR	100 (100)	97 (100,97)	91 (100,98)	712

trials by terms that are not patterns in the model. In this simulation, any pattern other than X_{50}, $X_{150}X_{250}$, or $X_{251}X_{252}$ is taken to be noise. For random forest, "noise" counts the total number of noisy variables selected in the 100 trials. Any variable other than the five in the logit function is noise.

On this example, pLPS selects all three patterns almost perfectly and generates the least amount of noise in the form of spuriously selected patterns. Logic Regression does not do well on the size-two patterns and selects slightly more noise. Random Forest does well in selecting the important variables but also selects many noisy variables. (If we change the criterion for declaring that Random Forest has selected a variable to the "top eight" or "top five," we reduce the number of noisy variables but also reduce the variable counts.) SPLR has similar performance to pLPS in selecting the patterns, but selects many more spurious patterns.

3.2 Simulation Example 3.2

Example 3.2 studies the behavior of pLPS on a large data set ($n = 1000$, $p = 8000$) with correlations among the covariates. To generate the binary variables X_i, $i = 1, 2, \ldots, p$, we start with normal distributions, choosing $X_i^* \sim N(0, 1)$, $i = 1, 2, \ldots, p$ so that $\text{corr}(X_i^*, X_{i+1}^*) =$

2/3 and $\text{corr}(X_i^*, X_{i+2}^*) = 1/3$, $i = 1, 2, \ldots, p-2$. (X_i^* and X_j^* are independent if $|i-j| > 2$.) We then set $X_i = 1$ if $X_i^* > 0$ and $X_i = 0$ otherwise, for each $i = 1, 2, \ldots, p$. The logit function is

$$f(x) = -4 + 2X_{500} + 3X_{5000} + 2X_{1000}X_{3000} + 3X_{7000}X_{7002}.$$

The simulation was repeated 50 times (each run is quite time-consuming). We could not run Logic Regression on this example, as the dimensions exceed the limit of that code.

Table 2 shows the results, in the same format as Table 1. pLPS misses the pattern $X_{1000}X_{3000}$ twice but selects the rest perfectly, and generates a smaller number of spurious noise patterns than the other methods. In Random Forest, we declared a variable to be selected if it was ranked in the top 12. It misses the pattern $X_{1000}X_{3000}$ with some frequency. SPLR selects all four patterns perfectly, but at the cost of a large number of spurious patterns. SPLR requires the user to set the maximum number of parameters allowed in the model, and selects the actual number by BIC. We set this maximum to 20, and it was reached on all 50 runs. (The maximum is still reached on every run when we set this parameter to 50.)

Table 2: Simulation Example 3.2 with $n = 1000$ and $p = 8000$, and correlations among neighboring variables. Tabulated numbers show the number of tests (out of 50) in which the pattern was detected by each algorithm. The number outside the parentheses is the number of times the given pattern was selected; the numbers inside the parentheses shows how many times the variables in the pattern are detected in the model, as a main effect or in some interaction. The final column shows the total number of times (in 50 tests) that the algorithms selected patterns (variables for RF) that do not appear in our model.

Methods	X_{500}	X_{5000}	$X_{1000}X_{3000}$	$X_{7000}X_{7002}$	noise
pLPS	50 (50)	50 (50)	48 (48,50)	50 (50,50)	278
RF	NA (50)	NA (50)	NA (28,37)	NA (50,50)	(335)
SPLR	50 (50)	50 (50)	50 (50,50)	50 (50,50)	800

3.3 Simulation Example 3.3

Example 3.3 studies the behavior of pLPS3 on a large data set, with sample size $n = 1000$ and $p = 500$ variables. The marginal distribution and correlation structure are the same as

in Example 3.2. The logit function is

$$f(x) = -4 + 2X_{100} + 3X_{200} + 2X_{300}X_{400} + 3X_{150}X_{450}X_{451}.$$

Again this simulation was repeated 50 times. As we can see from Table 3, pLPS3 selects all patterns quite well with a reasonable number of noisy patterns. Logic Regression selects fewer noisy patterns but does not do well in identifying the two interaction terms. Random Forest does well in the size-three pattern but misses the size two pattern quite often. (We declared the top 12 variables identified by Random Forest to be "selected"). As in the previous examples, SPLR does well at selecting the important patterns but also selects many noise patterns.

Table 3: Simulation Example 3.3, with $n = 1000$ and $p = 500$, and correlations among neighboring variables. Tabulated numbers show the number of tests (out of 50) in which the pattern was detected by each algorithm. The number outside the parentheses is the number of times the given pattern was selected; the numbers inside the parentheses shows how many times the variables in the pattern are detected in the model, as a main effect or in some interaction. The final column shows the total number of times (in 50 tests) that the algorithms selected patterns (variables for RF) that do not appear in our model.

Methods	X_{100}	X_{200}	$X_{300}X_{400}$	$X_{150}X_{450}X_{451}$	noise
pLPS3	47 (50)	50 (50)	47 (50,50)	47 (50,49,48)	204
Logic	50 (50)	50 (50)	34 (43,44)	30 (50,44,41)	151
RF	NA (50)	NA (50)	NA (36,40)	NA (49,47,49)	(279)
SPLR	50 (50)	50 (50)	45 (49,50)	50 (50,50,50)	554

To summarize the results obtained from simulated data: Logic Regression cannot handle very large data sets and does not reliably identify the interaction terms. Random Forest does not provide an explicit model of the interactions. It frequently scores well, but can perform poorly if the signal is not strong enough. SPLR scores well at selecting the right patterns, but selects too many noise patterns. By contrast, pLPS usually selects the right patterns without adding too many noise patterns.

4 The Gene Expression Barcode Data

With current microarray technology we are able to measure thousands of RNA transcripts at one time. This capability allows for richer characterization of cells and tissues. However, feature characteristics such as probe sequence can cause the observed intensity to be far away from the actual expression. Although the "probe effect" is large, it is consistent across different hybridizations, meaning that the effect is quite similar when comparing the intensities of different hybridizations for the same gene. Therefore, the majority of microarray data analysis uses relative expression rather than absolute expression. To overcome this limitation in measurement, a gene expression bar code (GEBC) [24] was proposed recently. The goal is to investigate what intensity measurement constitutes "no expression" for a given gene and microarray platform. GEBC starts by preprocessing all genes using Robust Multi-array Analysis (RMA) [5]. For each gene, an empirical density smoother is used to estimate the density function of this gene across tissues, and the smallest mode of the density function is taken to be the expected intensity of an unexpressed gene. Gene expressions to the left of this mode are used to estimate the standard deviation of unexpressed genes. If the log expression estimate of a gene is K standard deviations larger than the unexpressed mean, then this gene is considered to be expressed. The constant K is chosen to be 6 by cross-validation. For the purpose of our model, expressed genes are coded as 1 and unexpressed genes as 0.

GEBC [24] downloaded publicly available raw data from 40 different studies and created a database of 1094 human samples representing 118 different tissues. Of these samples, 503 are normal, 500 are breast tumors, and 91 are other diseases. A total of 22,215 genes are available for each sample.

We apply pLPS on this data set, with genes dichotomized by GEBC, as described above. Many genes have extremely unbalanced expression levels, being expressed (or unexpressed) in a very small percentage of the tissues. We removed these genes from our analysis, after which 7,654 genes remained. In our first analysis, we took all normal tissues as "controls" and all non-breast tumor tissues as "cases." In the second analysis we analyze the survival time of breast cancer patients after dichotomization. We define subjects with survival time less than 5 years as "cases" and those with survival time longer than 10 years as "controls."

4.1 Cancer

In this analysis, all normal and non-breast cancer tissues are used. Breast tumors were excluded because no normal breast tissue was available. The data set contains 503 normal tissues and 70 cancer tissues, giving a malignancy rate of 12.2%.

The model fitted by pLPS is shown in (10). Five size two interactions are selected.

$$\begin{aligned} f &= -8.15 + 3.58 \times CALU \times ERBB3 + 1.93 \times LAMC1 \times CD24 \\ &\quad +3.29 \times LPCAT1 \times ACY1 + 3.75 \times FXYD3 \times GNL3 \\ &\quad +2.34 \times NOTCH3 \times CD24. \end{aligned} \tag{10}$$

Most of these genes are known to be related to one or more types of cancer. For example, ERBB3 is very important in the development of breast cancer [16] and prostate cancer [8]. LPCAT1 is shown to be highly overexpressed in colorectal adenocarcinomas, when compared to normal mucosas [13]. ACY1 is found to be underexpressed in small-cell lung cancer (SCLC) cell lines and tumors [14]. FXYD3 is overexpressed in pancreatic ductal adenocarcinoma and influences pancreatic cancer cell growth [6]. Notch3 overexpression is common in pancreatic cancer [3]. Finally, CD24, one of the most well-known genes in this model, is related to breast cancer, ovarian cancer, NSCLC, and colorectal cancer [9] [10] [11] [22].

To compare the performance of pLPS with the alternative methods discussed in Section 3, the number of predictor genes must be reduced further, because Logic Regression cannot handle more than 1,000 variables. A screen step [18] was implemented to perform the reduction. We fitted a simple logistic regression on each gene and selected the most significant genes based on the p-values from the regression models. This step yields 636 genes.

We ran five-fold cross validation for all methods and summarized the results in Table 4. (Performance measures in this table are the average of the five-fold cross validation.) We tabulate the number of selected genes (# Gene), the number of non-zero coefficients (# Para), the highest order of interactions (q) and the summation of these three quantities (Total). The individual parameters measure the complexity of the model from different points of view, while the total provides an overall criterion. For prediction accuracy we present the area under the ROC curve in the column labelled "AUC". We can observe from these results that pLPS and pLPS3 select fewer genes; pLPS, pLPS3, and Logic use fewer

parameters than SPLR; pLPS and pLPS3 do not go to high order interactions because these are precluded by the model. In the total complexity criterion, there is a tie for first between pLPS and pLPS3. As for prediction, pLPS is the clear winner in AUC.

Table 4: Cancer data: Summary of results from five-fold cross validation. "Total" sums the number of selected genes, the number of non-zero coefficients in the model, and the highest order of interactions. AUC indicates the area under the ROC curve.

Methods	# Gene	# Para	q	Total	AUC
pLPS	9.2	6.6	**2.0**	**17.8**	**0.982**
pLPS3	**8.4**	6.4	3.0	**17.8**	0.945
Logic	14.0	**5.2**	5.0	24.2	0.956
SPLR	17.2	20.6	5.6	43.4	0.962

4.2 Breast Cancer Survival Time

The survival of breast cancer patients depends on many factors, such as grade, stage and oestrogen-receptor status. In this section we study the possible genetic effects using the gene expression barcode data. We denote patients who lived less than 5 years after diagnosis as "cases" and patients who lived more than 10 years after diagnosis as "controls." Patients with a censored death time less than 10 years and patients that died between 5 and 10 years are excluded. The remaining pool contains 243 patients, among which 80 are cases. The five-year death rate is $80/243 = 32.9\%$. As in the previous subsection, we used a screen step to reduce the size of the model. This step yielded 592 genes.

We applied the same methods with five-fold cross validation on the breast cancer survival data, summarizing the results in Table 5. Among the five measures presented, pLPS does the best in terms of the highest order of interactions and AUC measure, winning by a large margin over the other methods in the latter measure. Logic Regression performs surprisingly well in model complexity, selecting the smallest number of genes and parameters. However its prediction, as measured by AUC, has been sacrificed by the use of simple models.

(11) shows one model fitted by pLPS. There are one main effect and four size two interactions.

Table 5: Breast cancer survival data: Summary of results from five-fold cross validation. "Total" sums the number of selected genes, the number of non-zero coefficients, and the highest order of interactions. AUC indicates the area under the ROC curve.

Methods	# Gene	# Para	q	Total	AUC
pLPS	10.0	6.8	**2.0**	18.8	**0.824**
pLPS3	10.2	6.6	3.0	19.8	0.780
Logic	**4.4**	**2.6**	3.8	**10.8**	0.721
SPLR	19.4	20.6	5.0	45.0	0.793

$$\begin{aligned} f &= 3.21 - 1.59 \times PODXL - 2.00 \times SYNE2 \times AKAP11 + 2.05 \times CD20 \times CREB1 \\ &\quad -1.88 \times STAT5A \times MAPT - 1.89 \times MAOB \times IFFO1. \end{aligned} \tag{11}$$

Among these selected genes, CDC20, CREB1, STAT5A and MAPT are known to be related to breast cancer. It was noted in [7] that CDC20 is overexpressed in a large subset of malignancies such as colorectal, breast, lung and bladder cancers. The study [2] reports that CREB1 is much higher in breast tumor tissues as compared to non-neoplastic mammary tissues. Active STAT5 has been identified as a tumor marker of favorable prognosis in human breast cancer, and STAT5 activation is lost during metastatic progression [20]. It has been pointed out by [4] that MAPT inhibits the function of taxanes and high expression of MAPT decreased the sensitivity to taxanes.

5 Discussion

We have described a partitioned version of the LASSO-Patternsearch algorithm (named pLPS) that extends the range of this method to data sets with a higher number of predictors, and allows parallel execution of much of the computation. We show through simulations that pLPS is better than competing methods in selecting the correct variables and patterns while controlling for the amount of noise in the selected model. By testing on two gene expression data sets, we also show that pLPS gives smaller models with much better prediction accuracy than competing approaches.

Unlike LPS, two smoothing parameters with modified tuning criterion are used in pLPS and pLPS3. We impose a penalty on the difference between the number of main effects and the number of interactions for pLPS and a penalty on the difference among the numbers of main effects (size-two interactions in pLPS and size-three interactions in pLPS3). These penalties eliminate the extreme cases in which only main effects or interactions come up in the LASSO step. (These extreme cases appear too often with the original, unmodified criterion.) On the other hand, if an extreme case is the truth the LASSO step will generate some noisy patterns, but the parametric step tends to eliminate the noise and thus select the correct model.

References

[1] L. Breiman. Random forests. *Maching Learning*, 45:5–32, 2001.

[2] A. Chhabra, H. Fernando, G. Watkins, R. E. Mansel, and W. G. Jiang. Expression of transcription factor creb1 in human breast cancer and its correlation with prognosis. *Oncology Reports*, 18(4):953–958, 2007.

[3] T. Dang, K. Vo, K. Washington, and J. Berlin. The role of Notch3 signaling pathway in pancreatic cancer. *Journal of Clinical Oncology, 2007 ASCO Annual Meeting Proceedings, Part I*, 25(18S):21049, 2007.

[4] H. Ikeda, N. Taira, F. Hara, T. Fujita, H. Yamamoto, J. Soh, S. Toyooka, T. Nogami, T. Shien, H. Doihara, and S. Miyoshi. The estrogen receptor influences microtubule-associated protein tau (mapt) expression and the selective estrogen receptor inhibitor fulvestrant downregulates mapt and increases the sensitivity to taxane in breast cancer cells. *Breast Cancer Research*, 12:R43, 2010.

[5] R. A. Irizarry and B. Hobbs. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4:249–64, 2003.

[6] H. Kayed, J. Kleeff, A. Kolb, K. Ketterer, S. Keleg, K. Felix, T. Giese, R. Penzel, H. Zentgraf, M. W. Büchler, et al. FXYD3 is overexpressed in pancreatic ductal adenocarcinoma and influences pancreatic cancer cell growth. *International Journal of Cancer*, 118(1):43–54, 2006.

[7] T. Kidokoro, C. Tanikawa, Y. Furukawa, T. Katagiri, Y. Nakamura, and K. Matsuda. Cdc20, a potential cancer therapeutic target, is negatively regulated by p53. *Oncogene*, 27:1562–1571, 2008.

[8] I. H. Koumakpayi, J. S. Diallo, C. Le Page, L. Lessard, M. Gleave, L. R. Bégin, A. M. Mes-Masson, and F. Saad. Expression and nuclear localization of ErbB3 in prostate cancer. *Clinical Cancer Research*, 12(9):2730, 2006.

[9] G. Kristiansen, C. Denkert, K. Schluns, E. Dahl, C. Pilarsky, and S. Hauptmann. CD24 is expressed in ovarian cancer and is a new independent prognostic marker of patient survival. *American Journal of Pathology*, 161(4):1215, 2002.

[10] G. Kristiansen, K. Schlüns, Y. Yongwei, C. Denkert, M. Dietel, and I. Petersen. CD24 is an independent prognostic marker of survival in nonsmall cell lung cancer patients. *British Journal of Cancer*, 88(2):231–236, 2003.

[11] G. Kristiansen, K. J. Winzer, E. Mayordomo, J. Bellach, K. Schlüns, C. Denkert, E. Dahl, C. Pilarsky, P. Altevogt, H. Guski, et al. CD24 expression is a new prognostic marker in breast cancer. *Clinical Cancer Research*, 9(13):4906, 2003.

[12] LASSO-Patternsearch code. `http://pages.cs.wisc.edu/~swright/LPS/`.

[13] F. Mansilla, K. A. da Costa, S. Wang, M. Kruhøffer, T. M. Lewin, T. F. Ørntoft, R. A. Coleman, and K. Birkenkamp-Demtröder. Lysophosphatidylcholine acyltransferase 1 (LPCAT1) overexpression in human colorectal cancer. *Journal of Molecular Medicine*, 87(1):85–97, 2009.

[14] Y. E. Miller, J. D. Minna, and A. F. Gazdar. Lack of expression of aminoacylase-1 in small cell lung cancer. Evidence for inactivation of genes encoded by chromosome 3p. *Journal of Clinical Investigation*, 83(6):2120, 1989.

[15] M. Park and T. Hastie. Penalized logistic regression for detecting gene interactions. *Biostatistics*, 9(1)(30-50), 2008.

[16] E. Perez-Nadales and A. C. Lloyd. Essential function for ErbB3 in breast cancer proliferation. *Breast Cancer Research*, 6(3):137–139, 2004.

[17] I. Ruczinski, C. Kooperberg, and M. Leblanc. Logic regression. *Journal of Computational and Graphical Statistics*, 12:475–511, 2003.

[18] W. Shi, G. Wahba, and K. E. Lee. Detecting disease causing genes by LASSO-patternsearch algorithm. *Proceedings of BMC Genetics*, 2007.

[19] W. Shi, G. Wahba, S. J. Wright, K. E. Lee, R. Klein, and B. Klein. Lasso-patternsearch algorithm with applications to ophthalmology and genomic data. *Statistics and Its Interface*, 1:137–153, 2008.

[20] A. S. Sultan, J. Xie, M. J. LeBaron, E. L. Ealley, M. T. Nevalainen, and H. Rui. Stat5 promotes homotypic adhesion and inhibits invasive characteristics of human breast cancer cells. *Oncogene*, 24(5):746–60, 2005.

[21] W. Valdar, L. Solberg, D. Gauguier, S. Burnett, P. Klenerman, W. Cookson, M. Taylor, J. Rawlins, R. Mott, and J. Flint. Genome-wide genetic association of complex traits in heterogeneous stock mice. *Nature Genetics*, 38:879–887, 2006.

[22] W. Weichert, C. Denkert, M. Burkhardt, T. Gansukh, J. Bellach, P. Altevogt, M. Dietel, and G. Kristiansen. Cytoplasmic CD24 expression in colorectal cancer independently correlates with shortened patient survival. *Clinical Cancer Research*, 11(18):6574, 2005.

[23] S. J. Wright. Accelerated block-coordinate relaxation for regularized optimization. Technical report, University of Wisconsin-Madison, August 2010.

[24] M. J. Zilliox and R. A. Irizarry. A gene expression bar code for microarray data. *Nature Methods*, 4:911–913, 2007.