Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Uncorrelated Lasso

Si-Bao Chen
School of Computer Science and Technology, Anhui University, Hefei, 230601, China

Chris Ding
Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019, USA

Bin Luo and **Ying Xie**
School of Computer Science and Technology, Anhui University, Hefei, 230601, China

Abstract

Lasso-type variable selection has increasingly expanded its machine learning applications. In this paper, uncorrelated Lasso is proposed for variable selection, where variable de-correlation is considered simultaneously with variable selection, so that selected variables are uncorrelated as much as possible. An effective iterative algorithm, with the proof of convergence, is presented to solve the sparse optimization problem. Experiments on benchmark data sets show that the proposed method has better classification performance than many state-of-the-art variable selection methods.

In many regression applications, there are too many unrelated predictors which may hide the relationship between response and the most related predictors. A common way to resolve this problem is variable selection, that is to select a subset of the most representative or discriminative predictors from the input predictor set. The central requirement is that good predictor set contains predictors that are highly correlated with the response, but uncorrelated with each other. Various kinds of variable selection methods have been developed to tackle the issue of high dimensionality. The main challenge is to select a set of predictors, as small as possible, that help the classifier to accurately classify the learning examples.

The major type of variable selection methods (filter-type) is independent of classifiers, such as: t-test, F-statistic (Ding and Peng 2005), ReliefF (Kononenko 1994), mRMR (Peng, Long, and Ding 2005), and information gain/mutual information (Raileanu and Stoffel 2004). Another wrapper-type of variable selection methods take classifier as a black box to evaluate subsets of predictors (Kohavi and John 1997). There also is method of stochastic search for variable selection based on generalized singular g-prior (gsg-SSVS) (Yang and Song 2010).

Recently, sparsity regularization receives increasing attention in variable selection studies. The well-known Lasso (Least Absolute Shrinkage and Selection Operator) is a penalized least squares method with l_1-regularization, which is used to shrink/suppress variables to achieve the goal of variable selection (Tibshirani 1996). Owing to the nature of the l_1-norm penalty, the Lasso does both continuous shrinkage and automatic variable selection simultaneously. As variable selection becomes increasingly important in modern data analysis, the Lasso is much more appealing for its sparse representation. Elastic Net (Zou and Hastie 2005) added l_2-regularization in Lasso to make the regression coefficients more stable. Group Lasso (Yuan and Lin 2006) was proposed where the covariates are assumed to be clustered in groups, and the sum of Euclidean norms of the loadings in each group is utilized. Supervised Group Lasso (SGLasso) (Ma, Song, and Huang 2007) performed K-means clustering before Group Lasso.

In this paper, motivated by the previous sparse learning based research, we propose to add variable correlation into the sparse-learning-based variable selection approach. We note that in previous Lasso-type variable selection, variable correlations are not taken into account, while in most real-life data, predictors are often correlated. Strongly correlated predictors share similar properties, and have some overlapped information. In some cases, especially when the number of selected predictors is very limited, more information needs to be contained in the selected predictors, where strongly correlated predictors should not be in the model together. Only one predictor is selected out of the strongly correlated predictors, so that limited selected predictors will contain more information.Therefore we need to take into account the variable correlation in variable selection. To our knowledge, existing Lasso-type of variable selection methods have not considered variable correlation.

In the following, we firstly briefly review the normal Lasso and Elastic Net, then present our formulation of uncorrelated Lasso-type variable selection. An effective iterative algorithm, with its proof of convergence, is presented to solve the sparse optimization problem. Experiments on two benchmark gene data sets are performed to evaluate the algorithm. The paper concludes in the last section.

Brief review of Lasso and Elastic Net

Let there be a set of training data $\{(\mathbf{x}_i, y_i), i = 1, 2, \cdots, n\}$, where $\mathbf{x}_i = (x_{1i}, x_{2i}, \cdots, x_{pi})^\top \in \mathcal{R}^p$ is a vector of predictors and $y_i \in \mathcal{R}$ is its corresponding response. Formulate them in matrix form $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathcal{R}^{p\times n}$ and $\mathbf{y} = (y_1, y_2, \cdots, y_n)^\top \in \mathcal{R}^n$, then the Lasso (Tibshirani 1996) is a linear regression problem between predictors and

response, which can be written as

$$\min_{\beta \in \mathcal{R}^p} \|\mathbf{y}^\top - \beta^\top \mathbf{X}\|_2^2 + \lambda \|\beta\|_1, \tag{1}$$

where $\|\beta\|_1$ is l_1-norm of vector β (sum of absolute elements), $\|\beta\|_1 = \sum_{j=1}^{p} |\beta_j|$. $\lambda \geq 0$ is a tuning parameter. An intercept term is often omitted from (1) if the response and the predictors have been preprocessed by zero centering. The solution vector of (1) is very sparse (with few nonzero elements) due to the l_1-norm penalty. However, l_1-minimization algorithm is not stable compared with l_2-minimization (Xu, Caramanis, and Mannor 2012).

The Elastic Net (Zou and Hastie 2005) adds l_2-minimization term into Lasso objective function, which can be formulated as

$$\min_{\beta \in \mathcal{R}^p} \|\mathbf{y}^\top - \beta^\top \mathbf{X}\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \|\beta\|_2^2, \tag{2}$$

where $\lambda_1, \lambda_2 \geq 0$ are tuning parameters. Apart from enjoying a similar sparsity of representation of Lasso, the Elastic Net encourages a grouping effect, where strongly correlated predictors tend to be in or out of the model together (Zou and Hastie 2005).

Predictors with high correlation contain similar properties, and have some overlapped information. In some cases, especially when the number of selected predictors is very limited, more information needs to be contained in the selected predictors, where strongly correlated predictors should not be in the model together. Only one predictor is selected out of the strongly correlated predictors, so that limited selected predictors will contain more information.

Uncorrelated Lasso

In this section, we consider the variable selection based on Lasso-type l_1-minimization where selected predictors are uncorrelated as much as possible. Only one predictor of strongly correlated predictors tend to be in the model while the others not.

The Formulation

Suppose there are the matrix of predictors with n observations of p predictors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] = (x_{ki}) \in \mathcal{R}^{p \times n}$ and the corresponding response vector $\mathbf{y} = (y_1, y_2, \cdots, y_n)^\top \in \mathcal{R}^n$. Suppose response and all p predictors are preprocessed by normalization of zero mean and unit variance.

Denote the correlation coefficient matrix of p predictors $\mathbf{R} = (r_{kl}) \in [-1, 1]^{p \times p}$, where the (k, l)-th element r_{kl} is the correlation coefficient between the k and l-th zero-centered predictors,

$$r_{kl} = \frac{\sum_{i=1}^{n} x_{ki} x_{li}}{\sqrt{\sum_{i=1}^{n} x_{ki}^2} \sqrt{\sum_{i=1}^{n} x_{li}^2}}. \tag{3}$$

To let the selected predictors of Lasso-type l_1-minimization be uncorrelated as much as possible, the regression coefficient vector β should satisfy

$$\min_{\beta \in \mathcal{R}^p} \beta^\top \mathbf{C} \beta, \tag{4}$$

where

$$\mathbf{C} = \mathbf{R} \odot \mathbf{R} \tag{5}$$

is the square correlation coefficient matrix, $c_{kl} = r_{kl}^2$. $\odot$ is Hadamard product of matrices. We choose $\mathbf{C}$ instead of $\mathbf{R}$ to eliminate the effect of anti-correlation.

Therefore, we combine the above two minimization problem of Lasso (1) and decorrelation (4), and propose uncorrelated Lasso (ULasso) for representation and variable selection, which is formulated as

$$\min_{\beta \in \mathcal{R}^p} \|\mathbf{y}^\top - \beta^\top \mathbf{X}\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \beta^\top \mathbf{C} \beta, \tag{6}$$

where $\lambda_1, \lambda_2 \geq 0$ are tuning parameters. Note that correlation coefficient matrix $\mathbf{R}$ is semi-positive, then $\mathbf{C}$ is semi-positive. Therefore, Equation (6) is a convex optimization because all three terms are convex, which indicates that there exists a unique global optimum solution for minimizing (6) of ULasso.

By minimizing formula (6), one can obtain regression coefficients not only as sparse as that of Lasso but also with nonzero elements corresponding to predictors containing minimal correlations. If parameter λ_2 in (6) is set to be zero, then ULasso is reduced to normal Lasso. If the original predictors are all uncorrelated, i.e., $\mathbf{C} = \mathbf{I}_p$, then ULasso is turned into Elastic Net.

Optimization Algorithm

To obtain the global minimization solution of (6), we propose an iterative algorithm, which can be summarized as in Algorithm 1. In each iteration step, diagonal matrix $\mathbf{M}$ is calculated with the current β as in formula (7), and then β is updated based on the just calculated $\mathbf{M}$ as in formula (8). The iteration procedure between (7) and (8) is repeated until the algorithm converges.

Algorithm 1 Procedure of Uncorrelated Lasso

1: **Input:** Predictor matrix $\mathbf{X} \in \mathcal{R}^{p \times n}$ and corresponding response $\mathbf{y} \in R^n$ (response and all p predictors are zero-mean and unit variance), initial regression coefficients $\beta^{(0)} \in R^p$, tuning parameters $\lambda_1, \lambda_2 \geq 0$, maximum number of iteration t_{max} or residual bound $\epsilon > 0$;
2: Compute fixed matrix $\mathbf{B} = \mathbf{X}\mathbf{X}^\top + \lambda_2 \mathbf{C}$, $t = 0$;
3: Update diagonal matrix

$$\mathbf{M}^{(t)} = diag\left(\sqrt{|\beta_1^{(t)}|}, \sqrt{|\beta_2^{(t)}|}, \cdots, \sqrt{|\beta_p^{(t)}|}\right); \tag{7}$$

4: Update regression coefficients

$$\beta^{(t+1)} = \mathbf{M}^{(t)} \left[\mathbf{M}^{(t)} \mathbf{B} \mathbf{M}^{(t)} + \frac{\lambda_1}{2} \mathbf{I}_p\right]^{-1} \mathbf{M}^{(t)} \mathbf{X} \mathbf{y}; \tag{8}$$

5: If $t > t_{max}$ or $\|\beta^{(t+1)} - \beta^{(t)}\| < \epsilon$, go to step 6, otherwise, let $t = t + 1$ and go to step 3;
6: **Output:** The optimal regression coefficients $\beta^* = \beta^{(t+1)}$.

Note that in the input data of Algorithm 1, response and all p predictors are preprocessed by regularization of zero-

mean and unit variance. However, as discussed later, only the zero centering for each predictors is essential for Algorithm 1. Regularization of unit variance for predictors is used to balance among predictors which have different scales and variations, so that all predictors are treated equally when performing variable selection after Algorithm 1. Preprocessing of zero-mean and unit variance for response is optional, which is just adopted to simplify the prediction. In two-class case, decision bound of prediction can be simply set as zero.

Justification

In this section, we will see that Algorithm 1 does converge to the unique global optimum solution of ULasso minimization problem (6). Let $L(\beta)$ denote the objective function of ULasso in (6). Since $L(\beta)$ is a convex function of regression coefficients β, therefore, we only need to prove the objective function value $L(\beta)$ is non-increasing along each iterations in Algorithm 1, which is summarized in Theorem 1.

Theorem 1 *The objective function value $L(\beta)$ in ULasso minimization problem (6) is non-increasing, $L(\beta^{t+1}) \leq L(\beta^t)$, along with each iteration of formulae (7) and (8) in Algorithm 1.*

To prove the Theorem 1, we need the help of the following two Lemmas, which are needed to be proved firstly.

Lemma 2 *Define an auxiliary function*

$$G(\beta) = \|\mathbf{y}^\top - \beta^\top \mathbf{X}\|_2^2 + \lambda_1 \sum_{j=1}^{p} \frac{\beta_j^2}{2|\beta_j^{(t)}|} + \lambda_2 \beta^\top \mathbf{C}\beta. \quad (9)$$

Along with the $\{\beta^{(t)}, t = 0, 1, 2, \cdots\}$ sequence obtained in Algorithm 1, the following inequality holds,

$$G(\beta^{(t+1)}) \leq G(\beta^{(t)}). \quad (10)$$

Proof Since all three terms in auxiliary function $G(\beta)$ are semi-definite programming (SDP) problems, we can obtain the global optimal solution of $G(\beta)$ by taking the derivatives and let them equal to zero.

Making use of $\mathbf{M}^{(t)}$ denotation in (7), the auxiliary function $G(\beta)$ can be rewritten as

$$G(\beta) = \|\mathbf{y}^\top - \beta^\top \mathbf{X}\|_2^2 + \frac{\lambda_1}{2}\beta^\top (\mathbf{M}^{(t)})^{-2}\beta + \lambda_2 \beta^\top \mathbf{C}\beta. \quad (11)$$

Take the derivative of (11) with respect to β, and we get

$$\frac{\partial G(\beta)}{\partial \beta} = 2\mathbf{X}\mathbf{X}^\top \beta - 2\mathbf{X}\mathbf{y} + \frac{\lambda_1}{2} 2(\mathbf{M}^{(t)})^{-2}\beta + \lambda_2 2\mathbf{C}\beta. \quad (12)$$

By setting $\frac{\partial G(\beta)}{\partial \beta} = 0$, we obtain the optimal solution of auxiliary function

$$\begin{aligned} \beta^* &= \left[\mathbf{X}\mathbf{X}^\top + \frac{\lambda_1}{2}(\mathbf{M}^{(t)})^{-2} + \lambda_2 \mathbf{C}\right]^{-1} \mathbf{X}\mathbf{y} \quad (13) \\ &= \left[\mathbf{B} + \frac{\lambda_1}{2}(\mathbf{M}^{(t)})^{-2}\right]^{-1} \mathbf{X}\mathbf{y} \\ &= \mathbf{M}^{(t)} \left[\mathbf{M}^{(t)}\mathbf{B}\mathbf{M}^{(t)} + \frac{\lambda_1}{2}\mathbf{I}_p\right]^{-1} \mathbf{M}^{(t)}\mathbf{X}\mathbf{y}. \quad (14) \end{aligned}$$

The solution (14) gives the global optima of $G(\beta)$. Thus $G(\beta^*) \leq G(\beta)$ for any β. In particular, $G(\beta^*) \leq G(\beta^{(t)})$. Comparing (8) with (14), $\beta^{(t+1)} = \beta^*$. This completes the proof of Lemma 2.

It is important to note that we use Eq.(14) instead of the seemingly simpler Eq.(13). This is because as iteration progresses, some elements of β could become zero due to the sparsity of l_1-penalty. This causes the failure of inverse operator of $\mathbf{M}^{(t)}$ in Eq.(13). Thus Eq.(13) is ill defined. However, matrix $\mathbf{M}^{(t)}$ is well-defined. Thus Eq.(14) is well-defined, which is chosen as the updating rule (8) in Algorithm 1.

Lemma 3 *The $\{\beta^{(t)}, t = 0, 1, 2, \cdots\}$ sequence obtained by iteratively computing (7) and (8) in Algorithm 1 has the following property*

$$L(\beta^{(t+1)}) - L(\beta^{(t)}) \leq G(\beta^{(t+1)}) - G(\beta^{(t)}). \quad (15)$$

Proof Setting $\Delta = (L(\beta^{(t+1)}) - L(\beta^{(t)})) - (G(\beta^{(t+1)}) - G(\beta^{(t)}))$, substitute (6) and (9) in it,

$$\begin{aligned} \Delta &= (\lambda_1 \|\beta^{(t+1)}\|_1 - \lambda_1 \|\beta^{(t)}\|_1) - \\ &\quad \left(\lambda_1 \sum_{j=1}^{p} \frac{(\beta_j^{(t+1)})^2}{2|\beta_j^{(t)}|} - \lambda_1 \sum_{j=1}^{p} \frac{(\beta_j^{(t)})^2}{2|\beta_j^{(t)}|}\right) \\ &= -\frac{\lambda_1}{2} \sum_{j=1}^{p} \frac{1}{|\beta_j^{(t)}|} \{-2|\beta_j^{(t+1)}||\beta_j^{(t)}| + 2|\beta_j^{(t)}|^2 + \\ &\quad (\beta_j^{(t+1)})^2 - (\beta_j^{(t)})^2\} \\ &= -\frac{\lambda_1}{2} \sum_{j=1}^{p} \frac{1}{|\beta_j^{(t)}|} \left(|\beta_j^{(t+1)}| - |\beta_j^{(t)}|\right)^2 \\ &\leq 0. \quad (16) \end{aligned}$$

This completes the proof of Lemma 3.

From Lemma 2 and Lemma 3, we have,

$$L(\beta^{(t+1)}) - L(\beta^{(t)}) \leq G(\beta^{(t+1)}) - G(\beta^{(t)}) \leq 0, \quad (17)$$

which is to say

$$L(\beta^{(t+1)}) \leq L(\beta^{(t)}). \quad (18)$$

This completes the proof of Theorem 1. Therefore, Algorithm 1 converges to the global optimal solution of minimizing (6) of ULasso.

Variable Selection

When the global optimal solution $\beta^* = (\beta_1^*, \beta_2^*, \cdots, \beta_p^*)^\top$ of ULasso objective function (6) is obtain by Algorithm 1, small regression coefficients β_j^* are treated as zero for variable selection. Denote the filtered coefficients be $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2, \cdots, \hat{\beta}_p)^\top$,

$$\hat{\beta}_j = \begin{cases} \beta_j^*, & if |\beta_j^*| > \epsilon\mu \\ 0, & if |\beta_j^*| \leq \epsilon\mu \end{cases}, \quad (19)$$

where $\epsilon > 0$ is a small constant and $\mu = \sum_{j=1}^{p} |\beta_j^*|/p$ is absolute mean coefficient. Suppose there are k nonzero filtered

coefficients $|\hat{\beta}_{j_1}| \geq |\hat{\beta}_{j_2}| \geq \cdots \geq |\hat{\beta}_{j_k}| > 0$, then those predictors indexed by $\{j_1, j_2, \cdots, j_k\}$ are selected out.

In cases where different number of predictors need to be selected out, one should tune parameters λ_1 and λ_2 in objective function (6) to obtain different sparsity of filtered regression coefficients $\hat{\beta}$. A suboptimal but much simpler method of selecting q predictors is just picking out those predictors indexed by $\{j_1, j_2, \cdots, j_q\}$, which are corresponding to the q biggest absolute filtered regression coefficients $\{\hat{\beta}_{j_1}, \hat{\beta}_{j_2}, \cdots, \hat{\beta}_{j_q}\}$.

Prediction Bound: Two-Class Case

Suppose q predictors $\{x_{j_1}, x_{j_2}, \cdots, x_{j_q}\}$ are selected out, then the regression coefficients between q selected predictors and response needs to be recalculated with normal least-square method. Denote selected predictor vector $\mathbf{x}' = (x_{j_1}, x_{j_2}, \cdots, x_{j_q})^\top$, its corresponding observation matrix $\mathbf{X}' = [\mathbf{x}'_1, \mathbf{x}'_2, \cdots, \mathbf{x}'_n]$. The coefficients β' of regression equation $\mathbf{y}^\top = \beta'^\top \mathbf{X}'$ is re-estimated as

$$\hat{\beta}' = (\mathbf{X}'\mathbf{X}'^\top)^{-1}\mathbf{X}'\mathbf{y}. \tag{20}$$

The response of new observed predictor vector $\mathbf{x}' = (x_{j_1}, x_{j_2}, \cdots, x_{j_q})^\top$ can be predicted by

$$\hat{y} = \hat{\beta}'^\top \mathbf{x}'. \tag{21}$$

When response y is label information of two classes $y \in \{-1, 1\}$, the Bayesian optimal decision bound can be obtained as follows. Let all training samples of the two classes estimate their responses via formula (21), and denote the means and standard deviations of the estimated responses of the two classes be $\bar{y}_1$, $\bar{y}_2$ and σ_1, σ_2. Then, at Bayesian optimal decision bound b, the probability density of the two classes should be equal, $p(b|\bar{y}_1, \sigma_1) = p(b|\bar{y}_2, \sigma_2)$, which is equivalent to $(b - \bar{y}_1)/\sigma_1 = (\bar{y}_2 - b)/\sigma_2$. This gives the Bayesian optimal decision bound b,

$$b = \frac{\sigma_2 \bar{y}_1 + \sigma_1 \bar{y}_2}{\sigma_1 + \sigma_2}. \tag{22}$$

For new observed predictor vector $\mathbf{x}'$, its response $\hat{y}$ is predicted by (21). If $\hat{y} < b$, then make decision that $\mathbf{x}'$ belongs to class 1; otherwise, class 2.

Intercept Term

If predictors are not zero centered, then the intercept term t can not be omitted from the objective function (6) of ULasso. The formal objective function with intercept term t of ULasso, for any predictors $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] = (x_{ji}) \in \mathcal{R}^{p \times n}$ and response $\mathbf{y} = (y_1, y_2, \cdots, y_n)^\top \in \mathcal{R}^n$ without preprocessing, can be written as

$$\min_{\beta \in \mathcal{R}^p} \|\mathbf{y}^\top - t\mathbf{1}_n^\top - \beta^\top \mathbf{X}\|_2^2 + \lambda_1 \|\beta\|_1 + \lambda_2 \beta^\top \mathbf{C} \beta, \tag{23}$$

where $\mathbf{1}_n$ is an n-dimensional vector with all one entry.

Similar to previous method, an auxiliary function is constructed

$$G(t, \beta) = \|\mathbf{y}^\top - t\mathbf{1}_n^\top - \beta^\top \mathbf{X}\|_2^2 + \frac{\lambda_1}{2} \beta^\top \mathbf{M}^{-2} \beta + \lambda_2 \beta^\top \mathbf{C} \beta, \tag{24}$$

where diagonal matrix $\mathbf{M}$ is defined as in (7). Taking the derivative of $G(t, \beta)$ with respect to t and β, and letting them equal to zero, one can obtain

$$t = \frac{1}{n}(\mathbf{y}^\top - \beta^\top \mathbf{X})\mathbf{1}_n, \tag{25}$$

$$\beta = \left[\mathbf{X}\mathbf{X}^\top + \frac{\lambda_1}{2}\mathbf{M}^{-2} + \lambda_2 \mathbf{C}\right]^{-1} \mathbf{X}(\mathbf{y} - t\mathbf{1}_n). \tag{26}$$

From (25), we can get

$$\begin{aligned} \mathbf{X}(\mathbf{y} - t\mathbf{1}_n) &= \mathbf{X}(\mathbf{y} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top(\mathbf{y} - \mathbf{X}^\top\beta)) \\ &= \mathbf{X}((\mathbf{I}_n - \mathbf{P})\mathbf{y} + \mathbf{P}\mathbf{X}^\top\beta) \\ &= \tilde{\mathbf{X}}\mathbf{y} + \mathbf{X}\mathbf{P}\mathbf{X}^\top\beta, \end{aligned} \tag{27}$$

where $\mathbf{P} = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$, it satisfies $\mathbf{P}^2 = \mathbf{P}$ and $(\mathbf{I}_n - \mathbf{P})^2 = \mathbf{I}_n - \mathbf{P}$.

$$\tilde{\mathbf{X}} \hat{=} \mathbf{X}(\mathbf{I}_n - \mathbf{P}) = \mathbf{X} - \bar{\mathbf{x}}\mathbf{1}_n^\top \tag{28}$$

is the zero-centered predictor matrix.

Now replace (27) into (26), and we obtain

$$\begin{aligned} \beta^* &= \left[\mathbf{X}(\mathbf{I}_n - \mathbf{P})\mathbf{X}^\top + \frac{\lambda_1}{2}\mathbf{M}^{-2} + \lambda_2\mathbf{C}\right]^{-1}\tilde{\mathbf{X}}\mathbf{y} \\ &= \left[\tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top + \frac{\lambda_1}{2}\mathbf{M}^{-2} + \lambda_2\mathbf{C}\right]^{-1}\tilde{\mathbf{X}}\mathbf{y} \\ &= \mathbf{M}\left[\mathbf{M}\tilde{\mathbf{B}}\mathbf{M} + \frac{\lambda_1}{2}\mathbf{I}_p\right]^{-1}\mathbf{M}\tilde{\mathbf{X}}\mathbf{y}, \end{aligned} \tag{29}$$

where $\tilde{\mathbf{B}} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top + \lambda_2\mathbf{C}$. Comparing (29) with (8), we can see that the iteration procedure of ULasso with intercept term is similar to Algorithm 1, which needs to convert predictor matrix $\mathbf{X}$ to zero-centered $\tilde{\mathbf{X}}$ as (28) before iteration, then alternately update $\mathbf{M}$ and β as (7) and (29). As the iteration procedure converges, the intercept term t is estimated as (25), or

$$t^* = \bar{y} - \beta^{*\top}\bar{\mathbf{x}}, \tag{30}$$

where $\bar{y} = \sum_{i=1}^n y_i$ and $\bar{\mathbf{x}} = \sum_{i=1}^n \mathbf{x}_i$ are average response and average predictor vector of training samples.

Coefficient Initialization

From Theorem 1 we know that Algorithm 1 can converges to the global optimal solution from any nonzero initial coefficient $\beta^{(0)}$. However, different coefficient initialization may affect the convergence speed of Algorithm 1.

To evaluate the effect of different initial coefficient $\beta^{(0)}$ on the convergence speed of Algorithm 1, we design five initial coefficient $\beta^{(0)}$: first, all p entries are uniform random number between 0 and 1; second, all p entries are Gaussian random number of zero-mean and unit-variance; third, all p entries are equal to $\frac{1}{p}$; fourth, least square coefficient $\beta^{(0)} = (\mathbf{X}\mathbf{X}^\top + \lambda_2\mathbf{C})^{-1}\mathbf{X}\mathbf{y}$; and fifth, ridge regression coefficient $\beta^{(0)} = (\mathbf{X}\mathbf{X}^\top + \mathbf{I}_p)^{-1}\mathbf{X}\mathbf{y}$.

These five coefficient initializations are tested on Colon Cancer Data ((Alon et al. 1999)). Figure 1 shows the variation of objective function value along with each iteration

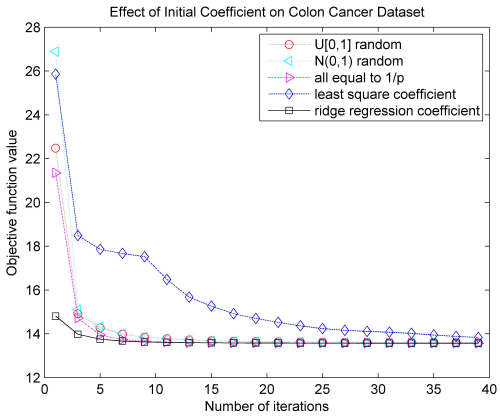

Figure 1: Effect of different initial regression coefficient.

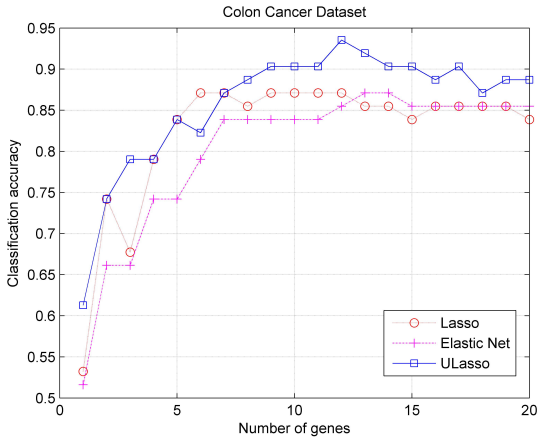

Figure 2: External LOOCV classification accuracy of ULasso on Colon Cancer Data.

steps when different initial coefficient is adopted in Algorithm 1. From the figure we can see that Algorithm 1 converges very quickly, whatever the initial coefficient. The lines corresponding to the first three initializations are very similar and converge almost the same time. Initialization with least square coefficient is a little slower than other initializations. The fastest convergence is corresponding to the initialization with ridge regression coefficient, since it is closer to the optimal solution.

Experiments

We evaluate the effectiveness of the proposed uncorrelated Lasso (ULasso) on two well known data sets: the Colon Cancer Data (Alon et al. 1999) and the Leukemia Dataset (Golub et al. 1999). The performance in variable selection and classification accuracy of the ULasso will be compared with other methods.

Colon Cancer Data

Alon et al. used Affymetrix Oligonucleotide Array to measure expression levels of 40 tumor and 22 normal colon tissues for 6500 human genes (Alon et al. 1999). These samples were collected from 40 different colon cancer patients, in which 22 patients supplied both normal and tumor samples. A subset of 2000 genes based on highest minimal intensity across the samples was selected, which can be downloaded from http:// microarray.princeton.edu/oncology/affydata/. These data are preprocessed by taking a base 10 logarithmic of each expression level, and then each tissue sample is standardized to zero mean and unit variance across the genes.

Since this dataset does not contain test set, leave-one-out cross validation (LOOCV) is usually adopted to evaluate the performance of the classification methods for a selected subset of genes. The external LOOCV procedure is performed as follows: 1) omit one observation of the training set; 2) based on the remaining observations, reduce the set of available genes to the top 200 genes as ranked in terms of the t statistic; 3) the q most significant genes were re-chosen from the 200 genes by the proposed ULasso algorithm; and 4) these q genes were used to classify the left out sample. This process was repeated for all observations in the training set until each observation had been picked out and classified exactly once.

Based on the LOOCV strategy, the classification performance of our method, with different q genes selected out, is plotted in Figure 2. From the figure, we can see that the performances of all three methods, Lasso, Elastic Net and ULasso, become better as more genes are picked out for classification. When the number of genes becomes large, the classification performances begin to saturate. When the number of genes is fixed, the performance of Elastic Net is comparable to Lasso. However, the proposed ULasso show consistent superiority over the Lasso and Elastic Net.

The top classification accuracy and the corresponding number of genes of the proposed ULasso are compared with the following classification methods: SVM (Furey et al. 2000), LogitBoost (Dettling and Bhlmann 2003), MAVE-LD (Antoniadis, Lambert-Lacroix, and Leblanc 2003), gsg-SSVS (Yang and Song 2010), Supervised group Lasso (SGLasso) (Ma, Song, and Huang 2007), Lasso (Tibshirani 1996) and Elastic Net (Zou and Hastie 2005). The summary is presented in Table 1. It is clear from the comparison that the proposed ULasso is better than the other popular classification methods using only moderate number of genes.

Method	No. of genes	LOOCV accuracy
SVM	1000 or 2000	0.9032
LogitBoost, optimal	2000	0.8710
MAVE-LD	50	0.8387
gsg-SSVS	10/14	0.8871
SGLasso	19	0.8710
Lasso	6	0.8710
Elastic Net	13	0.8710
ULasso	12	0.9355

Table 1: Top LOOCV accuracy and corresponding number of genes on Colon Cancer Data.

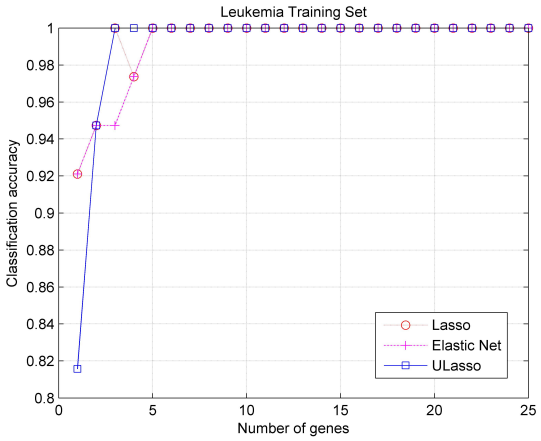

Figure 3: Classification accuracy of ULasso on Leukemia training set.

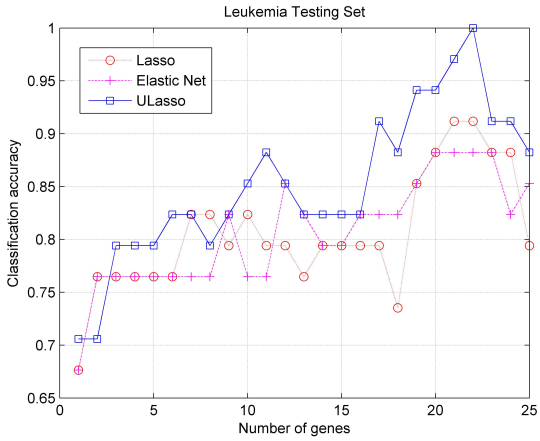

Figure 4: Classification accuracy of ULasso on Leukemia testing set.

Leukemia Dataset

The leukaemia data consist of 7129 genes and 72 samples (Golub et al. 1999), which can be downloaded from http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi. In the training data set, there are 38 samples, among which 27 are type 1 leukaemia (acute lymphoblastic leukaemia, ALL) and 11 are type 2 leukaemia (acute myeloid leukaemia, AML). The remaining 34 samples constitute test set, among which 20 are ALL and 14 are AML.

The preprocess method suggested by (Dudoit, Fridlyand, and Speed 2002) is taken for the data: 1) thresholding: floor of 100 and ceiling at $16,000$; 2) filtering: retain genes with $\max(gene)/\min(gene) > 5$ and $(\max(gene) - \min(gene)) > 500$, where $\max(gene)$ and $\min(gene)$ refer to the maximum and minimum expression levels of a particular gene across samples respectively; and 3) base 10 logarithmic transformation. The filtering resulted in 3571 genes. The gene expression data are further preprocessed to have mean zero and variance one across samples.

The gene selection procedure of our ULasso and other methods is trained on the training set. When a subset of genes are selected out for each methods, classification is performed both in training set and test set. Figure 3 and Figure 4 show the classification accuracy results of ULasso, compared with Lasso and Elastic Net, on Leukemia training and testing sets when different number of genes are selected. From the figures we can see that all of the three methods perform excellent on training set (all classify correct). On test set, Elastic Net is comparable to Lasso with the same number of genes selected out. Our proposed ULasso consistently outperforms the other two methods.

Then the top classification results of the proposed ULasso is compared with SVM (Furey et al. 2000), weighted voting machine (WVM) (Golub et al. 1999), MAVE-LD (Antoniadis, Lambert-Lacroix, and Leblanc 2003), gsg-SSVS (Yang and Song 2010), Lasso (Tibshirani 1996) and Elastic Net (Zou and Hastie 2005). The classification accuracy on training set and test set, and the corresponding number of genes are summarized in Table 2. From the table we can see that the proposed ULasso outperforms other methods with moderate number of genes.

Method	No. of genes	Training accuracy	Test accuracy
SVM	25∼2000	0.9474	0.8824∼0.9412
WVM	50	0.9474	0.8529
MAVE-LD	50	0.9737	0.9706
gsg-SSVS	14	0.9737	0.9706
Lasso	21	1.0000	0.9118
Elastic Net	26	1.0000	0.9118
ULasso	22	1.0000	1.0000

Table 2: Top accuracy and corresponding number of genes on Leukemia Dataset.

Conclusion

Lasso-type variable selection is learned with constrains of de-correlation, which is named uncorrelated Lasso (ULasso), so that the variables selected out are uncorrelated as much as possible. An effective iterative algorithm and its corresponding analysis, with proof of convergence, are proposed to solve ULasso. Experiments on two well known gene datasets show that the proposed ULasso has better classification performance than many state-of-the-art variable selection methods.

Acknowledgments

This work is supported by the Natural Science Foundation of China (61202228, 61073116), the Doctoral Program Foundation of Institutions of Higher Education of China (20103401120005), and Collegiate Natural Science Fund of Anhui Province (KJ2012A004).

References

Alon, U.; Barkai, N.; Notterman, D.; Gish, K.; Ybarra, S.; Mack, D.; and Levine, A. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences* 96(12):6745–6750.

Antoniadis, A.; Lambert-Lacroix, S.; and Leblanc, F. 2003. Effective dimension reduction methods for tumor classification using gene expression data. *Bioinformatics* 19(5):563–570.

Dettling, M., and Bhlmann, P. 2003. Boosting for tumor classification with gene expression data. *Bioinformatics* 19(9):1061–1069.

Ding, C. H. Q., and Peng, H. 2005. Minimum redundancy feature selection from microarray gene expression data. *J. Bioinformatics and Computational Biology* 3(2):185–206.

Dudoit, S.; Fridlyand, J.; and Speed, T. P. 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal Of The American Statistical Association* 97(457):77–87.

Furey, T. S.; Cristianini, N.; Duffy, N.; Bednarski, D. W.; Schummer, M.; and Haussler, D. 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *BMC Bioinformatics* 16(10):906–914.

Golub, T. R.; Slonim, D. K.; Tamayo, P.; Huard, C.; Gaasenbeek, M.; Mesirov, J. P.; Coller, H.; Loh, M. L.; Downing, J. R.; Caligiuri, M. A.; and et al. 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286(5439):531–537.

Kohavi, R., and John, G. H. 1997. Wrappers for feature subset selection. *Artif. Intell.* 97(1-2):273–324.

Kononenko, I. 1994. Estimating attributes: Analysis and extensions of RELIEF. In *European Conference on Machine Learning*, 171–182.

Ma, S.; Song, X.; and Huang, J. 2007. Supervised group lasso with applications to microarray data analysis. *BMC Bioinformatics* 8.

Peng, H.; Long, F.; and Ding, C. H. Q. 2005. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* 27(8):1226–1238.

Raileanu, L. E., and Stoffel, K. 2004. Theoretical comparison between the gini index and information gain criteria. *Ann. Math. Artif. Intell.* 41(1):77–93.

Tibshirani, R. 1996. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)* 58(1):267–288.

Xu, H.; Caramanis, C.; and Mannor, S. 2012. Sparse algorithms are not stable: A no-free-lunch theorem. *IEEE Trans. Pattern Anal. Mach. Intell.* 34(1):187–193.

Yang, A.-J., and Song, X.-Y. 2010. Bayesian variable selection for disease classification using gene expression data. *Bioinformatics* 26(2):215–222.

Yuan, M., and Lin, Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68(1):49–67.

Zou, H., and Hastie, T. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(2):301–320.