Expert Systems with Applications 38 (2011) 5110–5118

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Regularized logistic regression without a penalty term: An application to cancer classification with microarray data

Concha Bielza [a,*], Víctor Robles [b], Pedro Larrañaga [a]

[a] Department of Artificial Intelligence, Technical University of Madrid, Madrid, Spain
[b] Department of Computer Architecture and Technology, Technical University of Madrid, Madrid, Spain

ARTICLE INFO

Keywords:
Logistic regression
Regularization
Estimation of distribution algorithms
Cancer classification
Microarray data

ABSTRACT

Regularized logistic regression is a useful classification method for problems with few samples and a huge number of variables. This regression needs to determine the regularization term, which amounts to searching for the optimal penalty parameter and the norm of the regression coefficient vector. This paper presents a new regularized logistic regression method based on the evolution of the regression coefficients using estimation of distribution algorithms. The main novelty is that it avoids the determination of the regularization term. The chosen simulation method of new coefficients at each step of the evolutionary process guarantees their shrinkage as an intrinsic regularization. Experimental results comparing the behavior of the proposed method with Lasso and ridge logistic regression in three cancer classification problems with microarray data are shown.

1. Introduction

Logistic regression (Hosmer & Lemeshow, 2000) is a simple and efficient supervised classification method that provides explicit probabilities of class membership and an easy interpretation of the regression coefficients of predictor variables. The class variable is binary while the explanatory variables are of any type, not even requiring strong assumptions, like gaussianity of the predictor variables given the class or assumptions about the correlation structure. This lends great flexibility to this approach having shown a very good performance in a variety of fields (Baumgartner et al., 2004; Kiang, 2003).

Many of the most challenging current classification problems involve extremely high dimensionality k (thousands of variables) and small sample sizes N (less than one hundred cases). This is the so-called "large k, small N" problem, since it hinders proper parameter estimation when trying to build a classification model. Microarray data classification falls into this category.

In logistic regression we identify four problems in the "large k, small N" case. First, a large number of parameters – regression coefficients – have to be estimated using a very small number of samples. Therefore, an infinite number of solutions is possible as the problem is undetermined. Second, multicollinearity is largely present. As the dimensionality of the model increases, the chance grows that a variable can be constructed as a linear combination of other predictor variables, thereby supplying no new information. Third, over-fitting may occur, i.e. the model may fit the training data well but perform badly on new samples. These problems yield unstable parameter estimates. Fourth, there are also computational problems due to the large number of predictor variables. Traditional algorithms for finding the estimates numerically, like Newton–Raphson's method (Thisted, 1988), require prohibitive computations to invert a huge, sometimes singular matrix, at each iteration.

Within the context of logistic regression, the "large k, small N" problem has been tackled from three fronts: dimensionality reduction, feature (or variable) selection and regularization, or sometimes a combination of them.

As regards *dimensionality reduction*, principal components analysis is one of the most widespread methods (Aguilera, Escabias, & Valderrama, 2006). This preprocessing of high-dimensional variables outputs transformed variables, of which only a reduced set is used. These transformed variables are the classifier inputs. The main drawback is that principal components tend to need all the original variables in their expressions. As a result, the information requirements of model application are not reduced and there is also a loss of interpretability of the variables. Furthermore, there is not guarantee of class separability coinciding with the selected principal components (Weber, Vinterbo, & Ohno-Machado, 2004). Other methods, such as partial least squares (Antoniadis, Lambert-Lacroix, & Leblanc, 2003) or an adaptive dimension reduction through regression (Nguyen & Rocke, 2002) have also been used.

* Corresponding author. Tel.: +34 913366596; fax: +34 913524819.
E-mail addresses: mcbielza@fi.upm.es (C. Bielza), pedro.larranaga@fi.upm.es (V. Robles), vrobles@fi.upm.es (P. Larrañaga).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.09.140

Feature selection methods yield parsimonious models which reduce information costs, are easier to explain and understand, and increase model applicability and robustness. The selected features are good for discriminating between the different classes and may be sought via different heuristic search approaches (Liu & Motoda, 2008). The goodness of a proposed feature subset may be assessed via an initial screening process using a scoring metric. The metric is based on intrinsic characteristics of the data computed from simple statistics on the empirical distribution, totally ignoring the effects of the selected features on classifier performance. This is the so-called *filter* approach to feature selection in machine learning, or *screening* in statistics (West et al., 2001). By contrast, the *wrapper* approach searches good subsets using the classifier itself as part of their function evaluation (Kohavi & John, 1997). A performance estimate of the classifier trained with each subset assesses the merit of this subset. Some recent studies combine filter and wrapper approaches (Uncu & Türksen, 2007). In the context of logistic regression and $k \gg N$, Lee, Lee, Park, and Song (2005) propose different filter metrics to select a fixed number of features, the top-ranked ones, such that they are always fewer than the sample size.Avoiding the curse of dimensionality in a similar way, Weber et al. (2004) perform a preliminary feature selection by choosing the $N-1$ variables maximally correlated with the class variable. In a second phase, a logistic regression model is constructed with the selected features, and it is further simplified via a backwards variable selection.

The third front to tackle the "large k, small N" problem is using *regularization* methods. These methods impose a penalty on the size of logistic regression coefficients, trying to shrink them towards zero. Therefore, regularized estimators are restricted maximum likelihood estimators (MLE), since they maximize the likelihood function subject to restrictions on the logistic regression parameters. The little bias allowed provides more stable estimates with smaller variance. Regularization methods are more continuous than usual discrete processes of retaining-or-discarding features thereby not suffering as much from high variability (Hastie, Tibshirani, & Friedman, 2001). This shrinkage of coefficients was initially introduced in the ordinary linear regression scenario by Hoerl and Kennard (1970), where restrictions were spherical. This is the so-called ridge or quadratic (penalized) regression. Lee and Silvapulle (1988), LeCessie and vanHouwelingen (1992) extended the framework to logistic regression. Ridge estimators are expected to be on average closer to the real value of the parameters than the ordinary unrestricted MLEs, i.e. with smaller mean-squared error. See Fan and Li (2006), Bickel and Li (2006) for recent developments and a unified conceptual framework of the regularization theory.

Here we introduce *estimation of distribution algorithms* (EDAs) as intrinsic regularizers within the logistic regression context. EDAs are optimization heuristics included in the class of stochastic population-based search methods (Larrañaga & Lozano, 2002; Lozano, Larrañaga, Inza, & Bengoetxea, 2006; Pelikan, 2005). EDAs work by constructing an explicit probability model from a set of selected solutions, which is then conveniently used to generate new promising solutions in the next iteration of the evolutionary process. In our proposal, an EDA obtains the regularized estimates in a direct way in the sense that the objective function to be optimized is still the likelihood, not including any regularization term. It is a specifically chosen simulation process during the evolution which accounts intrinsically for the regularization. EDAs receive the unrestricted likelihood equations as inputs and generate the restricted MLEs as outputs.

The paper is organized as follows. Section 2 reviews both the classical and regularized versions of the logistic regression model. Section 3 describes EDAs and how we propose to use them to solve the regularized case. Experimental studies on several microarray data sets, a great exponent of the "large k, small N" problem, are presented in Section 4. Finally, Section 5 includes some conclusions and future work.

2. Regularized logistic regression

2.1. The need for regularizing logistic regression

Assume we have a (training) data set $\mathcal{D}_N$ of N independent samples from some experiment. $\mathcal{D}_N = \{(c_j, x_{j1}, \ldots, x_{jk}),\ j = 1, \ldots, N\}$, where $\mathbf{x}_j = (x_{j1}, \ldots, x_{jk})^t \in \mathbb{R}^k$ is the value of the jth sample, x_{ji} indicates the ith variable outcome of the jth sample and c_j is the known class label of the jth sample, 0 or 1, for the binary case considered in this paper.

Logistic regression uses the $\mathbf{x}$ values to determine the probability π of a sample belonging to one of the two classes. Thus, we have $k+1$ variables: the class or response dichotomous variable C and its predictor variables or covariates $X_1, \ldots, X_k$. The logistic model should be able to classify any new sample that comes along, characterized by just its covariate values.

Let π_j denote $P(C = 1|\mathbf{x}_j)$, $j = 1, \ldots, N$. Then the logistic regression model is defined as

$$\log \frac{\pi_j}{1-\pi_j} = \beta_0 + \sum_{i=1}^{k} \beta_i x_{ji} = \eta_j \Longleftrightarrow \pi_j = \frac{1}{1+e^{-\eta_j}} \tag{1}$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)^t$ denotes the vector of regression coefficients including a constant or intercept β_0. These are usually estimated from data by the maximum likelihood estimation method. From $\mathcal{D}_N$, the log-likelihood function is built as

$$l(\boldsymbol{\beta}) = \sum_{j=1}^{N} (c_j \log \pi_j + (1-c_j)\log(1-\pi_j)), \tag{2}$$

where π_j is given by expression (1). Maximum likelihood estimators, $\widehat{\beta}_i$, are obtained by maximizing l with respect to $\boldsymbol{\beta}$. Let $\mathbf{c}$ denote the vector of response values c_j ($j = 1, \ldots, N$), $\boldsymbol{\pi}$ be the vector of π_j values, $\mathbf{X}$ be an $N \times k$ matrix with each row given by $\mathbf{x}_j^t$, and $\mathbf{u}$ an N-vector of ones. Thus, the following system of $k+1$ equations and $k+1$ unknowns – called the likelihood equations – has to be solved:

$$\frac{\partial l}{\partial \boldsymbol{\beta}} = \mathbf{Z}^{\mathbf{t}}(\mathbf{c} - \boldsymbol{\pi}) = \mathbf{0},$$

where $\mathbf{Z}$ is the matrix $[\mathbf{u}|\mathbf{X}]$.

Newton–Raphson's algorithm is traditionally used to solve the resulting *nonlinear* equations for $\widehat{\beta}_i$ numerically. Each iteration provides an updating formula given by

$$\hat{\boldsymbol{\beta}}^{\text{new}} = \hat{\boldsymbol{\beta}}^{\text{old}} + (\mathbf{Z}^{\mathbf{t}}\mathbf{W}^{\mathbf{old}}\mathbf{Z})^{-1}\mathbf{Z}^{\mathbf{t}}(\mathbf{c} - \hat{\boldsymbol{\pi}}^{\text{old}}),$$

where $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)^t$, and $\hat{\boldsymbol{\pi}}$ denotes the vector of estimated values at that iteration, i.e. its jth-component is

$$\hat{\pi}_j^{\text{old}} = \left[1 + e^{-(\hat{\beta}_0^{\text{old}} + \hat{\beta}_1^{\text{old}} x_{j1} + \cdots + \hat{\beta}_{kx}^{\text{old}} x_{jk})}\right]^{-1}, \quad j = 1, \ldots, N$$

and $\mathbf{W}^{\text{old}}$ denotes a diagonal matrix with elements $\hat{\pi}_j^{\text{old}}\left(1 - \hat{\pi}_j^{\text{old}}\right)$.

In the context of data involving high dimensionality (k) and small sample sizes (N), the logistic regression approach has a number of problems, explained in the introduction section: undetermined problem to be solved, multicollinearity, over-fitting and computational difficulties. Regularization emerges as one of the most promising solutions for these problems. In this section we review the state-of-the-art in the case of regularized logistic regression.

Regularized logistic regression maximizes the penalized log-likelihood given by

$$l(\boldsymbol{\beta}) - \frac{\lambda}{2} J(\boldsymbol{\beta}), \tag{3}$$

where the penalty function is generally $J(\boldsymbol{\beta}) = \sum_i \gamma_i \psi(\beta_i), \gamma_i > 0$. Typical choices are $\psi(\beta_i) = |\beta_i|^q$, $q > 0$, and $\gamma_i = \gamma$, $\forall i$, giving rise to

$$l_q(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \frac{\lambda}{2} \sum_{i=1}^{k} |\beta_i|^q. \tag{4}$$

$\lambda > 0$ is the penalty or regularization parameter and controls the amount of shrinkage. The larger the λ, the stronger its influence is and the smaller the β_i sizes become. When $\lambda = 0$ the solution is the ordinary MLE, whereas if $\lambda \to \infty$, the β_i all tend to 0. λ is usually chosen by cross-validation. The cross-validated deviance, error, BIC or AIC are used as the criteria to be optimized.

2.2. Ridge logistic regression

The quadratically-regularized approach (i.e. $q = 2$), called *ridge* logistic regression, seeks MLEs subject to spherical restrictions on the parameters. Thus, the function to be maximized is

$$l_2(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \frac{\lambda}{2} \sum_{i=1}^{k} \beta_i^2. \tag{5}$$

The maximizer of $l_2(\boldsymbol{\beta})$ in expression (5) always exists and is unique. The objective function is smooth and concave, and as in the classical logistic regression, can be maximized by standard methods such as gradient descent, steepest descent, Newton, quasi-Newton, truncated Newton or conjugate-gradient.

From a Bayesian point of view, the ridge estimate is the posterior mode for a prior that is a flat prior for β_0 and independent distributions $N(0, \tau^2)$, where $\tau^2 = 1/\lambda$, for β_i (Hastie et al., 2001). Markov chain Monte Carlo techniques can be used, although the computational burden is very costly. The benefit is a better handling of model uncertainties.

In the field of microarray classification which is the most representative example of "the large k, small N" problem, literature on ridge logistic regression dates back to 2001 (Eilers, Boer, van Ommen, & van Houwelingen, 2001). However, even though Newton–Raphson's method simplifies the equations for obtaining the estimators in the same way as in classical logistic regression, we still have a computationally prohibitive problem in our "large k, small N" context: there are thousands of equations (in fact $(k + 1)$) to be solved, and the final equation given in Newton–Raphson's formulas requires a matrix of the same dimension to be inverted. Storing this information demands substantial memory space. Inverting huge matrices may be avoided to some extent with sophisticated algorithms, like the dual algorithm based on sequential minimal optimization (SMO) used in support vector machines and adapted in Keerthi, Duan, Shevade, and Poo (2005) to penalized logistic regression.

On the other hand, dimensionality reduction and feature selection techniques are again the solutions we find to avoid managing variables that are not discriminative between the classes and that degrade classifier performance. Thus, in the specific literature on DNA microarrays, (Shen & Tan, 2005) combine ridge logistic regression with partial least squares and with singular value decomposition (SVD), both of which are dimension-reduction methods. See Eilers et al. (2001), Hastie and Tibshirani (2004) for further details on efficient quadratic regularization for microarray data by using SVD. In addition, they use a feature selection method called recursive feature elimination (Guyon, Weston, Barnhill, & Vapnik, 2002) that iteratively removes genes with smaller absolute values of the ridge estimators. Similar ideas are explained in other works (Fort & Lambert-Lacroix, 2005; Nguyen & Rocke, 2002). In Zhu and Hastie (2004), in spite of reducing the matrix inversions required in the ridge logistic regression by using the SMO algorithm, generalized here to the multi-class case, the authors also apply several gene selection methods, including both filter and wrapper approaches. Estimating the classifier performance while ignoring the gene selection step can lead to severe downward bias. Liao and Chin (2007) propose a parametric bootstrap model for more accurate estimation of the performance.

2.3. Lasso logistic regression

When q is equal to 1 in $l_q(\boldsymbol{\beta})$ (see expression (4)) it results in *Lasso* (Least Absolute Shrinkage and Selection Operator), introduced by Tibshirani (1996) in the context of ordinary linear regression and later extended to logistic regression (Genkin, Lewis, & Madigan, 2007; Lokhorst, 1999; Shevade & Keerthi, 2003). The function to be maximized is

$$l_1(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \frac{\lambda}{2} \sum_{i=1}^{k} |\beta_i|, \tag{6}$$

Interest in Lasso is growing because its penalty encourages the estimators be either significantly large or exactly zero, which has the effect of automatically performing feature selection and hence yielding concise models.

In a Bayesian setting, the prior corresponding to this case is an independent Laplace distribution (or double exponential) for each β_i. Cawley and Talbot (2006) even model the penalty parameter λ by using a Jeffrey's prior to eliminate this parameter by integrating it out analytically.

Ng (2004) presents a theoretical result related to the sample complexity in the sense that the number of training examples required to learn "well" grows only logarithmically in the number of irrelevant features. Although the objective function is still concave in Lasso (as in ridge regression), an added computational problem is that this function is not differentiable. Generic methods for nondifferentiable concave problems, such as the ellipsoid method or subgradient methods, are usually very slow in practice. Faster methods have recently been investigated: interior point methods (Koh, Kim, & Boyd, 2007) and quadratic approximations to the likelihood function (Balakrishnan & Madigan, 2008; Lee, Lee, Abbeel, & Ng, 2006; Sha, Park, & Saul, 2007).

Besides the aforementioned cross-validated criteria for choosing λ, during the last years its determination has been carried out by using the regularization path that allows estimating $\boldsymbol{\beta}$ coefficients at the values of λ at which the (active) set of non-zero coefficients changes (Park & Hastie, 2006; Zhao & Yu, 2007).

Recent works propose variants of Lasso. Meier, van de Geer, and Bühlmann (2008) extend the *group* Lasso introduced by Yuan and Lin (2006) in the ordinary linear regression to logistic regression. Group Lasso is able to do variable selection on (predefined) groups of variables. The *fused* Lasso (Tibshirani, Saunders, Rosset, Zhu, & Knight, 2005) penalizes the $\boldsymbol{\beta}$ coefficients and their successive differences obtaining sparsity of both types of coefficients. The features are ordered in such a way as to make successive differences meaningful. Finally, the *adaptive* Lasso (Zou, 2006) uses adaptive weights for penalizing the coefficients differently.

2.4. Other regularizations

Bridge regression (Frank & Friedman, 1993) is the case with $q > 1$ in $\psi(\beta_i) = |\beta_i|^q$. Fu (1998) compares the bridge and Lasso in detail. With $q < 1$, the coefficients are more constrained than in Lasso leading to more sparse solutions. However, this formulation poses problems of nonconcavity and nondifferentiability and there is a lack of efficient computational methods (Liu et al., 2007). Recent works even advocate double penalizations: Lasso and ridge,

also called the elastic net penalty (Zhou & Hastie, 2005), or Firth's (Firth, 1993) and ridge (Gao & Shen, 2007).

Several authors find in DNA microarray classification an important field to apply regularized logistic regression. We have already mentioned some examples for ridge logistic regression in Section 2.2. Other regularizations also arise for microarrays: (Cawley & Talbot, 2006) is an example using Lasso and (Liu et al., 2007) is devoted to $q < 1$.

3. EDAs for regularizing logistic regression

Among the stochastic population-based search methods, EDAs (Larrañaga & Lozano, 2002; Lozano et al., 2006; Pelikan, 2005) have recently emerged as a general framework that overcomes some weaknesses of other well-known methods like genetic algorithms. Unlike genetic algorithms, EDAs avoid the ad hoc design of crossover and mutation operators, as well as the tuning of a large number of parameters, while they explicitly capture the relationships among the problem variables by means of a joint probability distribution (jpd). The main system underlying the EDA approach is:

(1) $D_0 \leftarrow$ Generate M individuals randomly. Evaluate them with a fitness function
(2) $h = 1$
(3) **do** {
(4) $D^{Se}_{h-1} \leftarrow$ Select $M' < M$ individuals from D_{h-1}
(5) $p_h(\mathbf{z}) = p\left(\mathbf{z}\middle|D^{Se}_{h-1}\right) \leftarrow$ Estimate the jpd from the selected individuals
(6) $D_h \leftarrow$ Sample M individuals (the new population) from $p_h(\mathbf{z})$ and evaluate them
(7) } **until** a stopping criterion is met

M individuals, each representing a point of the search space, constitute the initial population and are generated at random. All of them are evaluated by means of a fitness function (step 1). Then, M' ($M' < M$) individuals are selected according to a selection method, taking the fitness function into account (step 4). Next, a multidimensional probabilistic model that reflects the interdependencies between the encoded variables in these M' selected individuals is induced (step 5). The estimation of this underlying joint distribution represents the EDA bottleneck, as different degrees of complexity in the dependencies can be considered. In the next step, M new individuals – the new population – are obtained by sampling from the multidimensional probabilistic model learnt in the previous step (step 6). These steps, 4 to 6, are repeated until some pre-defined stopping condition is met (step 7).

If we confine ourselves to logistic regression classifiers, we find that other evolutionary algorithms like genetic algorithms have been used only for performing feature selection (Nakamichi, Imoto, & Miyano, 2004; Vinterbo & Ohno-Machado, 1999), but not for estimating the parameters.

As described above, regularized logistic regression may solve the problems encountered in "the large k, small N" context. The usual unrestricted MLEs are substituted by restricted MLEs that maximize a penalized likelihood. EDAs could be successfully used to optimize *any kind* of penalized likelihood, like the one in expression (3), because, unlike traditional numerical methods, they do not require derivative information or matrix inversions. EDAs would use expression (3) as the fitness function to guide the search while learning and simulating the distribution of the selected solutions. In this sense, EDAs would turn out to be a strong competitor of numerical methods.

Leaving aside this direct procedure, we investigate here a more interesting approach that shows that EDAs can act as an intrinsic regularizer if we choose a suitable representation. Thus, let us take $l(\boldsymbol{\beta})$ in expression (2) as the fitness function that assesses each possible solution $\boldsymbol{\beta}$ to the (unrestricted) maximum likelihood problem. $\boldsymbol{\beta}$ is a $k+1$ dimensional continuous random variable. EDAs would start by randomly generating the initial population D_0 of M individuals $\boldsymbol{\beta}_1^{(0)}, \ldots, \boldsymbol{\beta}_M^{(0)}$. After selecting M' individuals (e.g. the top M'), the core of the EDA paradigm is step 5 above to estimate the jpd from these selected M' individuals. Without losing generality, we start from a univariate marginal distribution algorithm $\left(\text{UMDA}_c^G\right)$ (Larrañaga, Etxeberria, Lozano, & Peña, 2000) in our continuous $\boldsymbol{\beta}$-domain. See González, Lozano, and Larrañaga (2002) for its theoretical support as an evolutionary algorithm to solve continuous optimization problems. UMDA_c^G assumes that at each generation h, all variables are independent and normally distributed, i.e.

$$p_h(\boldsymbol{\beta}) = \prod_{i=0}^{k} p_h(\beta_i) = \prod_{i=0}^{k} \frac{1}{\sigma_{ih}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\beta_i-\mu_{ih}}{\sigma_{ih}}\right)^2}. \quad (7)$$

We now modify UMDA_c^{G*} to tackle the regularized logistic regression by shrinking the β_i parameters during the EDA simulation step. The new algorithm is called UMDA_c^{G*}. Specifically, at step 5 UMDA_c^{G*} learns, at each iteration h, a model given by expression (7). This involves estimating the new μ_{ih} and σ_{ih} with the MLEs computed on the selected set D^{Se}_{h-1} of M' individuals from the previous generation. However, sampling at step 6 now generates individuals from (7) with the normal distributions $p_h(\beta_i)$ constrained to lie in an interval $[-b_h, b_h]$. This is readily achieved by generating values from a Gaussian of parameters μ_{ih} and σ_{ih} for each variable β_i and constraining its outputs, according to a standard rejection method – or via a transformation of that Gaussian – to fall within $[-b_h, b_h]$.

The idea is that, as long as the algorithm progresses, forcing the β_i parameters to be in a bounded interval around 0 constrains and stabilizes their values, just like regularization does. At step 5, we learn, for the random variable $\boldsymbol{\beta}$, the multivariate Gaussian distribution with a diagonal covariance matrix that best fits, in terms of likelihood, the M' $\boldsymbol{\beta}$-points that are top ranked in the objective function $l(\boldsymbol{\beta})$. We then generate, at step 6, M new points from the previous distribution truncated at each coordinate at $-b_h$ (bottom) and at b_h (top). New solutions are ranked with respect to their $l(\boldsymbol{\beta})$ values, and the best M' are chosen and so on. In spite of optimizing function $l(\boldsymbol{\beta})$ rather than another penalized log-likelihood function like $l_1(\boldsymbol{\beta})$ (in expression (6)) or $l_2(\boldsymbol{\beta})$ (in expression (5)), the evolutionary process guarantees that the β_i values belong to intervals of the desired size. Therefore, our estimates of β_i are regularized estimates. Moreover, since we use the original $l(\boldsymbol{\beta})$ objective function of the logistic regression, we do not need to specify the λ parameter of other penalized approaches like in expression (5).

Note that plenty of probability models are possible in expression (7), without necessarily assuming all variables to be Gaussian and independent. Different univariate, bivariate or multivariate dependencies may be designed with the benefit of having an explicit model of (possible) complex probabilistic relationships among the different parameters.

Finally, the last step, say at iteration $h = T$, would contain $\boldsymbol{\beta}_1^{(T)}, \ldots, \boldsymbol{\beta}_M^{(T)}$ from which $\text{argmax}_{j\in\{1,\ldots,M\}} l(\boldsymbol{\beta}_j^{(T)})$ would be chosen as the final regularized estimate of $\boldsymbol{\beta}$.

4. Results

We illustrate how our approach really acts as a regularizer on three publicly available[1] benchmark microarray data sets. First, the `Breast` data set (West et al., 2001) with 7129 genes and 49 tumor samples, 25 of them representing estrogen receptor-positive (ER+) and the other 24 being estrogen receptor-negative (ER-). Second, the `Colon` data set (Alon et al., 1999) that contains 2000

[1] http://bioinformatics.upmc.edu/Help/UPITTGED.html.

genes for 62 tissue samples: 40 cancer tissues and 22 normal tissues. Third, the `Leukemia` data set (Golub et al., 1999) that consists of 7129 genes and 72 tissue samples: 25 cases of acute myeloid leukemia (AML) and 47 cases of acute lymphoblastic leukemia (ALL).

For our proposal based on EDAs we have developed our own implementation in C++. The parameters used to run UMDA_c^{G*} were: an initial population of $M = 400$ individuals, $M' = 200$ selected individuals for learning and $b_h = 10$. The change in the mean fitness value between successive generations, i.e. in the mean value of the objective function $l(\boldsymbol{\beta})$, was the chosen criterion for assessing the convergence of the algorithms. The algorithm stops whenever this change is small enough so as not to detect improvement. Due to the stochastic nature of EDAs, each experiment is run ten times.

We compare our EDA with the most usual regularized versions, ridge and Lasso logistic regressions. The R environment (Ihaka & Gentleman, 1996) provides tested functions to obtain the estimates of $\boldsymbol{\beta}$ coefficients and of some classifier performance measures of interest. For ridge logistic regression, we use the `lrm()` R function from the `Design` package. For Lasso, we use the `gllce` function from the `lasso2` package. Using these functions we have adopted a simple scheme of searching the best λ along a grid of values with the error as the cross-validated criterion.

The classification accuracy or percentage of correctly classified observations is a typical performance measure to be maximized. However, this is not always a suitable metric specially when dealing with two-class problems with skewed classes and misclassification cost distributions. In this case, an effective and preferable criterion is the area under the receiver operating characteristic curve (AUC) (Hanley & McNeil, 1982). The AUC has a powerful interpretation and it is related to other well-known statistics making it easier to learn its statistical properties. The AUC ranges from 0 to 1, where perfect discrimination between both classes corresponds to an area of 1 (a horizontal line through the point (1,1)) and random classification corresponds to an area of 0.5 (the identity line).

Demšar (2006) finds astounding that classification accuracy is usually still the only measure used, despite the medical and machine learning communities urge us to use other measures like AUC. Cortes and Mohri (2004), Huang and Ling (2005) studied in detail the relationship between classification accuracy and AUC and concluded that, although both measures reveal separate characteristics of a classifier, the AUC is statistically consistent and a more discriminating measure than classification accuracy. Moreover, the AUC is also a suitable measure to assess the classifier ability to rank instances in two-class classification problems. In particular, the AUC is the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. A review of different ways to estimate the AUC, both parametric and non-parametric, may be found in Lasko, Bhagwat, Zou, and Ohno-Machado (2005). Therefore, in this paper we will record both measures, accuracy and AUC. We use the `somers2()` R function included in the `Hmisc` package to estimate the AUC by means of the c-index.

Thus, our aim is to compare our EDA-based algorithm, UMDA_c^{G*}, as an intrinsic regularizer, against other well-known regularized logistic regressions: Ridge and Lasso. Starting from the same fixed set of genes, each of the three algorithms – UMDA_c^{G*}, Ridge and Lasso – constructs the logistic classifier by estimating the parameters according to its own methodology. The benefit of combining a regularization with a dimension-reduction step to enhance classifier efficiency has been pointed out elsewhere (Fort & Lambert-Lacroix, 2005). This preliminary selection of genes is based on different filter metrics usually found in the literature. We have used four filter criteria: (1) the BSS/WSS criterion, which maximizes the ratio of between-class to within-class sums of squares (as in Dudoit, Fridlyand, & Speed (2002)), (2) a ranking of genes according to their Pearson correlation coefficient to the class variable (as in Weber et al. (2004), West et al. (2001)), (3) a *p-metric* that looks for genes with maximum difference between the two within-class mean expression levels (as in Inza, Larrañaga, Blanco, & Cerrolaza (2004)), and (4) a *t-score* based on a statistical standard t-test.

Table 1
Results of UMDA_c^{G*} vs. other logistic regressions for `Breast` with the BSS/WSS criterion. ● and ♦ symbols are used for the comparisons UMDA_c^{G*} vs. Ridge and UMDA_c^{G*} vs. Lasso, respectively. ▼ means that Ridge or Lasso is statistically superior to UMDA_c^{G*} (p-value < 0.05).

# genes	UMDA_c^{G*}		Ridge		Lasso	
	Accur.	AUC	Accur.	AUC	Accur.	AUC
1	0.8613	0.9426	0.8643	0.9405	0.8593	0.9416
2	0.8421	0.9266 ♦	0.8557 ▼	0.9310 ▼	0.8517 ▼	0.9235
3	0.9077 ♦	0.9577 ● ♦	0.9128 ▼	0.9545	0.8560	0.9390
4	0.9062 ♦	0.9600 ● ♦	0.9208 ▼	0.9557	0.8605	0.9396
5	0.8921 ♦	0.9504 ● ♦	0.9104 ▼	0.9442	0.8587	0.9328
6	0.9167 ♦	0.9743 ● ♦	0.9334 ▼	0.9678	0.8605	0.9398
7	0.9103 ♦	0.9727 ● ♦	0.9292 ▼	0.9665	0.8614	0.9395
8	0.9249 ♦	0.9818 ● ♦	0.9410 ▼	0.9771	0.8604	0.9421
9	0.9179 ♦	0.9794 ● ♦	0.9357 ▼	0.9753	0.8597	0.9430
10	0.9213 ♦	0.9850 ● ♦	0.9375 ▼	0.9747	0.8698	0.9400
11	0.9307 ♦	0.9889 ● ♦	0.9422 ▼	0.9787	0.8712	0.9422
12	0.9224 ♦	0.9862 ● ♦	0.9348 ▼	0.9768	0.8749	0.9427
13	0.9245 ♦	0.9844 ● ♦	0.9348 ▼	0.9760	0.8729	0.9406
14	0.9224 ♦	0.9861 ● ♦	0.9333 ▼	0.9753	0.8711	0.9393
15	0.9257 ♦	0.9875 ● ♦	0.9310 ▼	0.9734	0.8719	0.9402
16	0.9258 ♦	0.9869 ● ♦	0.9289 ▼	0.9729	0.8715	0.9389
17	0.9207 ♦	0.9836 ● ♦	0.9267 ▼	0.9693	0.8715	0.9386
18	0.9156 ♦	0.9819 ● ♦	0.9252 ▼	0.9692	0.8679	0.9378
19	0.9165 ♦	0.9802 ● ♦	0.9236 ▼	0.9649	0.8684	0.9368
20	0.9287 ♦	0.9835 ● ♦	0.9334 ▼	0.9729	0.8675	0.9382
21	0.9309 ♦	0.9855 ● ♦	0.9362 ▼	0.9768	0.8668	0.9366
22	0.9327 ♦	0.9850 ● ♦	0.9378 ▼	0.9780	0.8660	0.9360
23	0.9371 ♦	0.9856 ● ♦	0.9413 ▼	0.9780	0.8643	0.9369
24	0.9355 ♦	0.9847 ● ♦	0.9408 ▼	0.9779	0.8636	0.9356
25	0.9333 ♦	0.9847 ● ♦	0.9380 ▼	0.9768	0.8645	0.9356
26	0.9340 ♦	0.9843 ♦			0.8631	0.9355
27	0.9337 ♦	0.9839 ♦			0.8621	0.9346
28	0.9329 ♦	0.9834 ♦			0.8648	0.9351
29	0.9325 ♦	0.9833 ♦			0.8606	0.9320
30	0.9345 ♦	0.9839 ♦			0.8619	0.9344
31	0.9326 ♦	0.9820 ♦			0.8630	0.9343
32	0.9305 ♦	0.9811 ♦			0.8613	0.9329
33	0.9310 ♦	0.9810 ♦			0.8631	0.9320
34	0.9315 ♦	0.9813 ♦			0.8597	0.9326
35	0.9301 ♦	0.9807 ♦			0.8606	0.9342
36	0.9377 ♦	0.9828 ♦			0.8593	0.9341
37	0.9363 ♦	0.9822 ♦			0.8607	0.9328
38	0.9360 ♦	0.9817 ♦			0.8585	0.9315
39	0.9379 ♦	0.9837 ♦			0.8585	0.9330
40	0.9416 ♦	0.9865 ♦			0.8558	0.9309
41	0.9422 ♦	0.9862 ♦			0.8583	0.9310
42	0.9400 ♦	0.9850 ♦			0.8583	0.9311
43	0.9395 ♦	0.9857 ♦			0.8566	0.9291
44	0.9382 ♦	0.9847 ♦			0.8562	0.9300
45	0.9396 ♦	0.9851 ♦			0.8593	0.9315
46	0.9386 ♦	0.9844 ♦			0.8564	0.9311
47	0.9408 ♦	0.9847 ♦			0.8542	0.9286
48	0.9384 ♦	0.9842 ♦			0.8562	0.9299
49	0.9489 ♦	0.9894 ♦			0.8538	0.9291
50	0.9502 ♦	0.9889 ♦			0.8545	0.9297

The method for estimating the classifier's performance measures should be carefully chosen. In our case, these measures are classification accuracy and AUC. The *holdout* estimation method is impractical with small samples. *Cross-validation* estimation provides unreliable estimates for small samples due to excessive variance, which is problematic in microarray analysis. The behavior of cross-validation for very small samples has been thoroughly studied in Braga-Neto and Dougherty (2004) who did not even find substantial differences, in terms of decreased variance, among the cross-validated variants (leave-one-out, 5- and 10-fold, stratified and repeated cross-validation).A large variance is of particular

Table 2
Results of UMDA$_c^{G*}$ vs. other logistic regressions for `Colon` with the BSS/WSS criterion. ● and ♦ symbols are used for the comparisons UMDA$_c^{G*}$ vs. RIDGE and UMDA$_c^{G*}$ vs. LASSO, respectively. ▼ means that RIDGE or LASSO is statistically superior to UMDA$_c^{G*}$ (p-value < 0.05).

# genes	UMDA$_c^{G*}$		RIDGE		LASSO	
	Accur.	AUC	Accur.	AUC	Accur.	AUC
1	0.8423 ♦	0.8188	0.8487	0.8213	0.8016	0.8208
2	0.8233 ♦	0.8163	0.8255	0.8224 ▼	0.8142	0.8223 ▼
3	0.8204 ♦	0.8507 ♦	0.8277 ▼	0.8591 ▼	0.7965	0.8400
4	0.8151 ♦	0.8442 ♦	0.8205 ▼	0.8443	0.7956	0.8314
5	0.8348 ♦	0.8316 ● ♦	0.8470 ▼	0.8230	0.7927	0.8290
6	0.8406 ♦	0.8365 ♦	0.8624 ▼	0.8466 ▼	0.7907	0.8274
7	0.8521 ♦	0.8436	0.8629 ▼	0.8484 ▼	0.8014	0.8424
8	0.8449 ♦	0.8345	0.8634 ▼	0.8485 ▼	0.8006	0.8423 ▼
9	0.8409 ♦	0.8256	0.8520 ▼	0.8361 ▼	0.7993	0.8407 ▼
10	0.8589 ♦	0.9219 ♦	0.9097 ▼	0.9219	0.7883	0.8981
11	0.8663 ♦	0.9248 ● ♦	0.9155 ▼	0.9180	0.7864	0.8960
12	0.8678 ♦	0.9224 ● ♦	0.9123 ▼	0.9179	0.7842	0.8928
13	0.8625 ♦	0.9235 ● ♦	0.9059 ▼	0.9087	0.7848	0.8932
14	0.8853 ♦	0.9477 ● ♦	0.9126 ▼	0.9261	0.7817	0.8980
15	0.8899 ♦	0.9479 ● ♦	0.9158 ▼	0.9305	0.7833	0.8963
16	0.8859 ♦	0.9453 ● ♦	0.9115 ▼	0.9272	0.7839	0.8944
17	0.8985 ♦	0.9595 ● ♦	0.9167 ▼	0.9387	0.7845	0.8957
18	0.8967 ♦	0.9601 ● ♦	0.9154 ▼	0.9413	0.7844	0.8948
19	0.8939 ♦	0.9547 ● ♦	0.9123 ▼	0.9374	0.7846	0.8969
20	0.8921 ♦	0.9522 ● ♦	0.9080 ▼	0.9357	0.7841	0.8983
21	0.8869 ♦	0.9490 ● ♦	0.9031 ▼	0.9299	0.7841	0.8937
22	0.8870 ♦	0.9507 ● ♦	0.9050 ▼	0.9338	0.7828	0.8942
23	0.8853 ♦	0.9488 ● ♦	0.9023 ▼	0.9317	0.7828	0.8942
24	0.8871 ♦	0.9481 ● ♦	0.9017 ▼	0.9283	0.7842	0.8949
25	0.8855 ♦	0.9456 ● ♦	0.9010 ▼	0.9283	0.7832	0.8930
26	0.8826 ♦	0.9424 ● ♦	0.8983 ▼	0.9227	0.7831	0.8919
27	0.8832 ♦	0.9433 ● ♦	0.9025 ▼	0.9308	0.7824	0.8936
28	0.8808 ♦	0.9404 ● ♦	0.8990 ▼	0.9287	0.7836	0.8949
29	0.8779 ♦	0.9395 ● ♦	0.8980 ▼	0.9274	0.7806	0.8933
30	0.8738 ♦	0.9363 ● ♦	0.8942 ▼	0.9256	0.7829	0.8941
31	0.8785 ♦	0.9352 ● ♦	0.8955 ▼	0.9237	0.7816	0.8934
32	0.8744 ♦	0.9302 ● ♦	0.8924 ▼	0.9205	0.7821	0.8932
33	0.8830 ♦	0.9395 ● ♦	0.8996 ▼	0.9315	0.7821	0.8928
34	0.8807 ♦	0.9371 ● ♦	0.8994 ▼	0.9322	0.7809	0.8921
35	0.8790 ♦	0.9355 ● ♦	0.9015 ▼	0.9330	0.7797	0.8961
36	0.8798 ♦	0.9362 ♦			0.7808	0.8947
37	0.8800 ♦	0.9366 ♦			0.7795	0.8960
38	0.8776 ♦	0.9347 ♦			0.7779	0.8937
39	0.8763 ♦	0.9332 ♦			0.7795	0.8926
40	0.8781 ♦	0.9369 ♦			0.7794	0.8951
41	0.8794 ♦	0.9339 ♦			0.7785	0.8928
42	0.8767 ♦	0.9326 ♦			0.7789	0.8930
43	0.8807 ♦	0.9361 ♦			0.7774	0.8916
44	0.8778 ♦	0.9330 ♦			0.7794	0.8916
45	0.8795 ♦	0.9336 ♦			0.7781	0.8922
46	0.8821 ♦	0.9365 ♦			0.7782	0.8919
47	0.8805 ♦	0.9351 ♦			0.7793	0.8928
48	0.8779 ♦	0.9328 ♦			0.7804	0.8923
49	0.8798 ♦	0.9353 ♦			0.7770	0.8905
50	0.8795 ♦	0.9343 ♦			0.7797	0.8912

Table 3
Results of UMDA$_c^{G*}$ vs. other logistic regressions for `Leukemia` with the BSS/WSS criterion. ● and ♦ symbols are used for the comparisons UMDA$_c^{G*}$ vs. RIDGE and UMDA$_c^{G*}$ vs. LASSO, respectively. ▼ means that RIDGE or LASSO is statistically superior to UMDA$_c^{G*}$ (p-value < 0.05).

# genes	UMDA$_c^{G*}$		RIDGE		LASSO	
	Accur.	AUC	Accur.	AUC	Accur.	AUC
1	0.9326 ♦	0.9793 ●	0.9362 ▼	0.9784	0.8550	0.9786
2	0.9228 ♦	0.9815 ♦	0.9246 ▼	0.9810	0.8631	0.9750
3	0.9445 ♦	0.9934 ♦	0.9559 ▼	0.9933	0.8518	0.9796
4	0.9441 ♦	0.9920 ● ♦	0.9578 ▼	0.9893	0.8525	0.9816
5	0.9378 ♦	0.9870 ● ♦	0.9494 ▼	0.9831	0.8504	0.9805
6	0.9342 ♦	0.9817 ● ♦	0.9481 ▼	0.9766	0.8580	0.9769
7	0.9370 ♦	0.9838 ● ♦	0.9544 ▼	0.9783	0.8547	0.9768
8	0.9325 ♦	0.9817 ● ♦	0.9548 ▼	0.9775	0.8536	0.9763
9	0.9382 ♦	0.9801 ● ♦	0.9553 ▼	0.9750	0.8529	0.9759
10	0.9383 ♦	0.9772 ● ♦	0.9546 ▼	0.9713	0.8529	0.9756
11	0.9455 ♦	0.9792 ● ♦	0.9587 ▼	0.9761	0.8496	0.9747
12	0.9459 ♦	0.9730 ●	0.9553 ▼	0.9667	0.8488	0.9730
13	0.9423 ♦	0.9674 ●	0.9537 ▼	0.9617	0.8467	0.9714 ▼
14	0.9450 ♦	0.9662 ●	0.9541 ▼	0.9594	0.8459	0.9711 ▼
15	0.9444 ♦	0.9693 ●	0.9577 ▼	0.9677	0.8469	0.9713
16	0.9448 ♦	0.9716 ● ♦	0.9556 ▼	0.9659	0.8459	0.9689
17	0.9465 ♦	0.9647 ●	0.9551 ▼	0.9619	0.8445	0.9680 ▼
18	0.9441 ♦	0.9657	0.9562 ▼	0.9666	0.8456	0.9691 ▼
19	0.9489 ♦	0.9627	0.9568 ▼	0.9644	0.8436	0.9681 ▼
20	0.9506 ♦	0.9652	0.9617 ▼	0.9685 ▼	0.8354	0.9698 ▼
21	0.9579 ♦	0.9743 ♦	0.9649 ▼	0.9751	0.8356	0.9716
22	0.9557 ♦	0.9738 ♦	0.9663 ▼	0.9775 ▼	0.8347	0.9711
23	0.9576 ♦	0.9738 ♦	0.9663 ▼	0.9782 ▼	0.8368	0.9712
24	0.9611 ♦	0.9795 ♦	0.9700 ▼	0.9812 ▼	0.8352	0.9713
25	0.9626 ♦	0.9832 ♦	0.9716 ▼	0.9831	0.8367	0.9717
26	0.9630 ♦	0.9860 ● ♦	0.9729 ▼	0.9842	0.8363	0.9717
27	0.9631 ♦	0.9853 ● ♦	0.9730 ▼	0.9826	0.8361	0.9718
28	0.9627 ♦	0.9849 ● ♦	0.9728 ▼	0.9833	0.8356	0.9711
29	0.9623 ♦	0.9848 ● ♦	0.9717 ▼	0.9831	0.8358	0.9710
30	0.9609 ♦	0.9846 ● ♦	0.9714 ▼	0.9827	0.8349	0.9712
31	0.9618 ♦	0.9843 ● ♦	0.9707 ▼	0.9822	0.8360	0.9711
32	0.9626 ♦	0.9867 ● ♦	0.9719 ▼	0.9844	0.8358	0.9730
33	0.9622 ♦	0.9859 ● ♦	0.9725 ▼	0.9841	0.8337	0.9704
34	0.9676 ♦	0.9894 ♦	0.9795 ▼	0.9921 ▼	0.8313	0.9718
35	0.9665 ♦	0.9897 ♦	0.9781 ▼	0.9918 ▼	0.8325	0.9726
36	0.9725 ♦	0.9926 ♦	0.9842 ▼	0.9955 ▼	0.8320	0.9722
37	0.9713 ♦	0.9923 ♦	0.9837 ▼	0.9950 ▼	0.8313	0.9719
38	0.9702 ♦	0.9920 ♦	0.9831 ▼	0.9952 ▼	0.8322	0.9723
39	0.9698 ♦	0.9918 ♦	0.9814 ▼	0.9945 ▼	0.8322	0.9724
40	0.9703 ♦	0.9914 ♦	0.9824 ▼	0.9948 ▼	0.8328	0.9720
41	0.9712 ♦	0.9918 ♦			0.8324	0.9724
42	0.9713 ♦	0.9927 ♦			0.8326	0.9735
43	0.9711 ♦	0.9914 ♦			0.8314	0.9716
44	0.9725 ♦	0.9904 ♦			0.8323	0.9720
45	0.9761 ♦	0.9925 ♦			0.8319	0.9716
46	0.9770 ♦	0.9923 ♦			0.8326	0.9721
47	0.9778 ♦	0.9938 ♦			0.8326	0.9720
48	0.9753 ♦	0.9926 ♦			0.8316	0.9728
49	0.9757 ♦	0.9924 ♦			0.8306	0.9713
50	0.9756 ♦	0.9925 ♦			0.8298	0.9716

concern in our small sample case since the estimate can often be far from the actual performance measure. Bootstrap estimation procedures are smoothed versions of cross-validation to reduce the variability of performance estimates. They come at the price of a high computational cost and an increased bias. Braga-Neto and Dougherty (2004) proved the .632 *bootstrap* estimator (Efron, 1983) to be a good overall estimator in small-sample microarray classification, and it is therefore the chosen method in this paper. Based on our experience, a good choice in the experiments for the number B of bootstrap samples used for training is $B = 500$. Note that for each bootstrap sample, the search for λ must be carried out, thereby increasing the computational burden.

Tables 1–3 summarize the experimental results of the mean performance measures, accuracy and AUC, over the ten executions, once a fixed number of genes {1,2,3,...,50} has been selected and scored by the four different filter procedures. Due to space limitations, we only show the BSS/WSS filter per data set. The results and

Table 4
Some statistical measures of the run times (in seconds).

		min	mean	max
`Breast`	UMDA$_c^{G*}$	0.31	1.76	3.98
	RIDGE	0.28	0.65	1.08
	LASSO	0.27	0.45	0.68
`Colon`	UMDA$_c^{G*}$	0.91	8.45	14.69
	RIDGE	0.27	0.78	1.20
	LASSO	0.26	0.53	0.75
`Leukemia`	UMDA$_c^{G*}$	0.35	3.80	7.92
	RIDGE	0.18	0.59	0.99
	LASSO	0.18	0.39	0.67

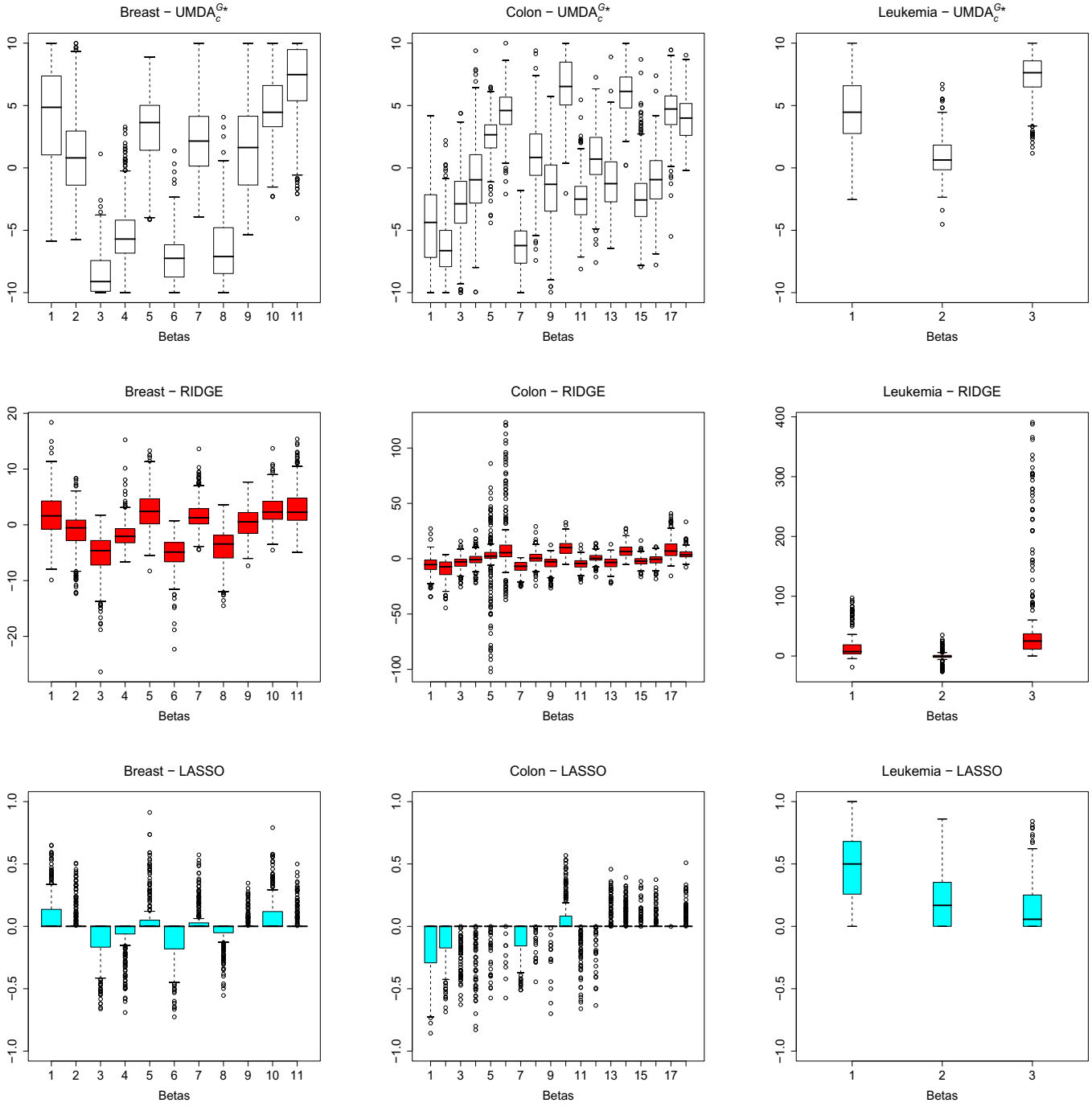

Fig. 1. Boxplots for each β_i for `Breast` with 11 genes (left), `Colon` with 18 genes (center) and `Leukemia` with 3 genes (right), with UMDA$_c^{G*}$ (top), RIDGE (center) and LASSO (bottom) algorithms.

tables of the remaining filters are available on our web page.[2] We have to remark that the influence of the filters is not so sizeable. The Mann–Whitney test was used to compute the statistical significance of the difference between a pair of algorithms: We tested both whether UMDA$_c^{G*}$ exhibits a statistically significant better behavior than the other algorithm and vice versa, i.e. whether the other algorithm reveals a statistically significant better behavior than UMDA$_c^{G*}$. For comparing UMDA$_c^{G*}$ with RIDGE and LASSO, the symbols used are ● and ♦, respectively. The symbols mean that UMDA$^{G*}{}_c$ reveals a statistically significant better behavior when compared to the other algorithm with respect to the performance measure, accuracy or AUC, depending on the column, with a p-value < 0.05. When RIDGE or LASSO is statistically superior to UMDA$_c^{G*}$, the symbol used is ▼, meaning that UMDA$_c^{G*}$ is beaten.

The conclusions are as follows. First, when compared to LASSO, UMDA$_c^{G*}$ is statistically superior both on AUC and accuracy measures for all data sets (see ♦ symbols in columns 2 and 3). For some isolated cases, LASSO is superior to UMDA$^{G*}{}_c$: for `Breast`, with 2 genes on accuracy (see column 6 in Table 1); for `Colon`, with 2, 8 and 9 genes on AUC (see column 7 in Table 2); and for `Leukemia`, with 13, 14 and 17 to 20 genes on AUC (see column 7 in Table 3).

Second, when compared to RIDGE, UMDA$_c^{G*}$ is statistically superior on the AUC measure for all data sets for almost any number

[2] http://laurel.datsi.fi.upm.es/~vrobles/reg_eda.

of genes (see ● symbols in column 3 of the three tables). In some isolated cases, the test provides statistically significant difference in favor of RIDGE (see ▼ symbols in column 5 of the three tables). However, RIDGE is statistically superior to UMDA$_c^{G*}$ on the accuracy measure (see ▼ symbols in column 4).

Although not displayed in the tables (see our web page), as expected, UMDA$_c^{G*}$ always exhibits a statistically significant superiority both on AUC and on accuracy against the classical logistic regression.

Note the blank results for RIDGE in the lower part of the tables. The regression does not work with more than 25 genes for `Breast`, 35 for `Colon` and 40 for `Leukemia`. This number is related to the average $0.632N$ of the original data points that are expected to be obtained (not repeated) in the bootstrap sample of size N. Likewise LASSO, UMDA$_c^{G*}$ and EDAs in general, offer an attractive alternative as they do not have this limitation.

As far as computational burden is concerned, UMDA$_c^{G*}$ is costlier than RIDGE and LASSO, the latter being the fastest algorithm. Table 4 shows a summary of the run times for the three methods. Despite being slower, UMDA$^{G*}{}_c$ yields rather acceptable times, ranging from less than 1 CPU second running on an Intel Xeon 2 GHz under Linux to almost 4 s for `Breast`, almost 8 s for `Leukemia` and almost 15 s for `Colon`.

The rationale behind this behavior may be the following. LASSO indeed works with fewer variables than the other methods because it yields a sparser vector $\boldsymbol{\beta}$, with relatively few nonzero coefficients.Thus, for example, when 20 genes are pre-selected, LASSO is actually working with 11 genes for `Breast` (there are 9 zero coefficients) and 6 genes for `Colon` and `Leukemia` (14 coefficients are zero), see our web page. On the contrary, RIDGE and UMDA$_c^{G*}$ typically yield $\boldsymbol{\beta}$s with *all* coefficients nonzero.

On the other hand, RIDGE and LASSO have to search a good λ value whereas UMDA$_c^{G*}$ does not. UMDA$^{G*}{}_c$, however, explores and evaluates more possible solutions than the other algorithms and has the additional steps of learning and simulation. With regard to the objective function to be maximized, UMDA$_c^{G*}$'s is simpler than RIDGE's, which despite being differentiable has the penalty term, and than LASSO's, which in addition is non-differentiable.

By further analyzing the results of the Tables 1–3 we can suggest a good model for UMDA$_c^{G*}$. It would be desirable to have good performance measures, accuracy and AUC, but also with a reasonable number of genes. Thus, the model with 11 genes seems to be the most suitable for `Breast`, with an accuracy of 0.9307 and AUC equal to 0.9889; for `Colon`, it is the model with 18 genes with an accuracy of 0.8967 and AUC equal to 0.9601; whereas for `Leukemia`, only 3 genes make up a good choice, with an accuracy of 0.9445 and AUC equal to 0.9934.

Interestingly enough, we show how our method is a regularizer since the β_i estimates are indeed stable. Fig. 1 shows the boxplots of the 10×500 bootstrap estimates of β_i coefficients for the models marked above as suitable: for `Breast` with 11 genes, `Colon` with 18 genes and `Leukemia` with 3 genes. Note that the Y-axis scales are different depending on the algorithm and the data set.

UMDA$_c^{G*}$ behaves as expected from a good regularizer: β_i's variability is low and there are only a few outliers in all the estimates. Although exhibiting more outliers than UMDA$^{G*}{}_c$, LASSO is indeed the algorithm that shrinks the β_i estimates the most. However, RIDGE is the worst method in this regard. Note that the same pattern of the boxes is always reproduced regardless of the algorithm.

5. Conclusion and future work

We have introduced a novel EDA-based approach that finds a regularized logistic classifier. EDA is not influenced by situations where the number of covariates is relatively large compared to the number of observations. By including the shrinkage of the coefficients intrinsically during its evolution process while optimizing the usual likelihood function, our approach works like a regularized logistic classifier. EDAs receive the unrestricted likelihood equations as inputs and generate the restricted MLEs as outputs.

Our proposal yields significantly better performance on the relevant AUC measure, as compared to ridge and Lasso logistic regressions. The classification accuracy achieved outperforms that of Lasso although it is worse than the accuracy obtained with ridge logistic regression. Our evolutionary strategy takes longer to find the coefficient estimates, ridge and Lasso logistic regressions being faster. However, run times are still negligible. Finally, we have shown our regularization to be effective on the stability of the regression parameter estimates. Therefore, the intrinsic regularizer presented here turns up as a good candidate in the regularized logistic regression context.

Future directions to be explored are EDA approaches that take into account more complex probabilistic conditional dependencies among β_i parameters, at the expense, perhaps, of a higher computational cost. Traditional numerical methods are unable to provide this kind of information. The inclusion of interaction terms among (possibly co-regulated) genes in η_j of Eq. (1) would also be feasible.

Finally, unlike the traditional numerical procedures, the EDA approach could be used in a more direct way, as a method that is able to optimize *any* objective function, regardless of its complexity or the non-existence of an explicit formula for its expression. Thus, EDA could find parameters that maximize any regularized logistic regression (Lasso, bridge...) or even the AUC objective. The difficulty in dealing with the AUC directly as the objective function is pointed out in Ma and Huang (2005), who use an approximation to it instead. Nevertheless, it is the original and intrinsic way of shrinking the regression coefficients embedded in some EDA steps which provides our valuable contribution in this paper.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Education and Science, projects TIN2007-62626, TIN2007-67148, TIN2005-03824 and Consolider Ingenio 2010-CSD2007-00018, and by the National Institutes of Health (USA), project 1 R01 LM009520-01.

References

Aguilera, A. M., Escabias, M., & Valderrama, M. J. (2006). Using principal components for estimating logistic regression with high-dimensional multicollinear data. *Computational Statistics and Data Analysis, 50*, 1905–1924.

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

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

Balakrishnan, S., & Madigan, D. (2008). Algorithms for sparse linear classifier in the massive data setting. *Journal of Machine Learning Research, 9*, 313–337.

Baumgartner, C., Böhm, C., Baumgartner, D., Marini, G., Weinberger, K., Olgemöller, B., et al. (2004). Supervised machine learning techniques for the classification of metabolic disorders in newborns. *Bioinformatics, 20*, 2985–2996.

Bickel, P. J., & Li, B. (2006). Regularization in statistics. *Test, 15*, 271–344.

Braga-Neto, U. M., & Dougherty, E. R. (2004). Is cross-validation valid for small-sample microarray classification? *Bioinformatics, 20*, 374–380.

Cawley, G. C., & Talbot, N. (2006). Gene selection in cancer classification using sparse logistic regression with Bayesian regularization. *Bioinformatics, 22*, 2348–2355.

Cortes, C., & Mohri, M. (2004). *AUC optimization vs. error rate minimization. Advances in neural information processing systems* (Vol. 16). Cambridge, MA: The MIT Press.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research, 7*, 1–30.

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

Efron, B. (1983). Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association, 78*, 316–331.

Eilers, P., Boer, J., van Ommen, G., & van Houwelingen, H. (2001). Classification of microarray data with penalized logistic regression. *Proceedings of SPIE. Progress in Biomedical Optics and Images, 4266*(2), 187–198.

Fan, J., & Li, R. (2006). Statistical challenges with high dimensionality: Feature selection in knowledge discovery. In *Proceedings of the Madrid international congress of mathematicians* (Vol. III, pp. 595–622).

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika, 80*, 27–38.

Fort, G., & Lambert-Lacroix, S. (2005). Classification using partial least squares with penalized logistic regression. *Bioinformatics, 21*, 1104–1111.

Frank, I. E., & Friedman, J. H. (1993). A statistical view of some chemometric regression tools. *Technometrics, 35*, 109–148.

Fu, W. J. (1998). Penalized regression: The bridge versus the LASSO. *Journal of Computational and Graphical Statistics, 7*, 397–416.

Gao, S., & Shen, J. (2007). Asymptotic properties of a double penalized maximum likelihood estimator in logistic regression. *Statistics and Probability Letters, 77*, 925–930.

Genkin, A., Lewis, D. D., & Madigan, D. (2007). Large-scale Bayesian logistic regression for text categorization. *Technometrics, 49*, 291–304.

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

González, C., Lozano, J. A., & Larrañaga, P. (2002). Mathematical modelling of $UMDA_c$ algorithm with tournament selection. Behaviour on linear and quadratic functions. *International Journal of Approximate Reasoning, 31*, 313–340.

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning, 46*, 389–422.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 143*, 29–36.

Hastie, T., & Tibshirani, R. (2004). Efficient quadratic regularization for expression arrays. *Biostatistics, 5*, 329–340.

Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning*. Springer.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimates for nonorthogonal problems. *Technometrics, 12*, 55–67.

Hosmer, D. W., & Lemeshow, S. (2000). *Applied logistic regression* (2nd ed.). New York: J. Wiley and Sons.

Huang, J., & Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering, 17*, 299–310.

Ihaka, R., & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics, 5*, 229–314.

Inza, I., Larrañaga, P., Blanco, R., & Cerrolaza, A. (2004). Filter versus wrapper gene selection approaches in DNA microarray domains. *Artificial Intelligence in Medicine, 31*, 91–103.

Keerthi, S. S., Duan, K. B., Shevade, S. K., & Poo, A. N. (2005). A fast dual algorithm for kernel logistic regression. *Machine Learning, 61*, 151–165.

Kiang, M. Y. (2003). A comparative assessment of classification methods. *Decision Support Systems, 35*, 441–454.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*, 273–324.

Koh, K., Kim, S.-Y., & Boyd, S. (2007). An interior-point method for large-scale l_1-regularized logistic regression. *Journal of Machine Learning Research, 8*, 1519–1555.

Larrañaga, P., Etxeberria, R., Lozano, J. A., & Peña, J. M. (2000). Optimization in continuous domains by learning and simulation of Gaussian networks. In *Workshop in optimization by building and using probabilistic models. Genetic and evolutionary computation conference, GECCO 2000* (pp. 201–204).

Larrañaga, P., & Lozano, J. A. (Eds.). (2002). *Estimation of distribution algorithms. A new tool for evolutionary computation*. Kluwer A.P.

Lasko, T. A., Bhagwat, J. G., Zou, K. H., & Ohno-Machado, L. (2005). The use of ROC curves in biomedical informatics. *Journal of Biomedical Informatics, 38*, 404–415.

Le Cessie, S., & van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Applied Statistics, 41*, 191–201.

Lee, S.-I., Lee, H., Abbeel, P., & Ng, A. Y. (2006). Efficient L1 regularized logistic regression. *Proceedings of the 21st national conference on artificial intelligence (AAAI-06)* (pp. 1–9).

Lee, J. W., Lee, J. B., Park, M., & Song, S. H. (2005). An extensive comparison of recent classification tools applied to microarray data. *Computational Statistics and Data Analysis, 48*, 869–885.

Lee, A., & Silvapulle, M. (1988). Ridge estimation in logistic regression. *Communications in Statistics, Part B–Simulation and Computation, 17*, 1231–1257.

Liao, J. G., & Chin, K.-V. (2007). Logistic regression for disease classification using microarray data: Model selection in a large p and small n case. *Bioinformatics, 23*, 1945–1951.

Liu, Z., Jiang, F., Tian, G., Wang, S., Sato, F., Meltzer, S. J., et al. (2007). Sparse logistic regression with L_p penalty for biomarker identification. *Statistical Applications in Genetics and Molecular Biology, 6* [Article 6].

Liu, H., & Motoda, H. (2008). *Computational methods of feature selection*. Chapman and Hall/CRC Press.

Lokhorst, J. (1999) The lasso and generalized linear models. Technical Report, University of Adelaide.

Lozano, J. A., Larrañaga, P., Inza, I., & Bengoetxea, E. (Eds.). (2006). *Towards a new evolutionary computation. Advances in estimation of distribution algorithms*. New York: Springer.

Ma, S., & Huang, J. (2005). Regularized ROC method for disease classification and biomarker selection with microarray data. *Bioinformatics, 21*, 4356–4362.

Meier, L., van de Geer, S., & Bühlmann, P. (2008). The group Lasso for logistic regression. *Journal of the Royal Statistical Society, Series B, 70*, 53–71.

Nakamichi, R. E., Imoto, S., & Miyano, S. (2004). Case-control study of binary disease trait considering interactions between SNPs and environmental effects using logistic regression. In *Fourth IEEE symposium on bioinformatics and bioengineering* (Vol. 21, pp. 73–78).

Ng, A. (2004). Feature selection, L_1 vs. L_2 regularization, and rotational invariants. In *Proceedings of the 21st international conference on machine learning*.

Nguyen, D. V., & Rocke, D. M. (2002). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics, 18*, 39–50.

Park, M. Y., & Hastie, T. (2006). L_1 regularization path algorithm for generalized linear models. Technical Report, Stanford University.

Pelikan, M. (2005). *Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms*. Springer.

Sha, F., Park, Y. A., & Saul, L. K. (2007). *Multiplicative updates for L_1-regularized linear and logistic regression. Lecture notes in computer science* (Vol. 4723). Springer.

Shen, L., & Tan, E. C. (2005). Dimension reduction-based penalized logistic regression for cancer classification using microarray data. *IEEE Transactions on Computational Biology and Bioinformatics, 2*, 166–175.

Shevade, S. K., & Keerthi, S. S. (2003). A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics, 19*, 2246–2253.

Thisted, R. A. (1988). *Elements of statistical computing*. New York: Chapman and Hall.

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

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smothness via the fused lasso. *Journal of the Royal Statistical Society, Series B, 67*, 91–108.

Uncu, O., & Türksen, I. B. (2007). A novel feature selection approach: Combining feature wrappers and filters. *Information Sciences, 177*, 449–466.

Vinterbo, S., & Ohno-Machado, L. (1999). A genetic algorithm to select variables in logistic regression: Example in the domain of myocardial infarct. *Journal of the American Medical Informatics Association, 6*, 984–988.

Weber, G., Vinterbo, S., & Ohno-Machado, L. (2004). Multivariate selection of genetic markers in diagnostic classification. *Artificial Intelligence in Medicine, 31*, 155–167.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., et al. (2001). Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences USA, 98*(20), 11462–11467.

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

Zhao, P., & Yu, B. (2007). Stagewise Lasso. *Journal of Machine Learning Research, 8*, 2701–2726.

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

Zhu, J., & Hastie, T. (2004). Classification of gene microarrays by penalized logistic regression. *Biostatistics, 5*, 427–443.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association, 101*, 1418–1429.