Penalized Logistic Regression and Classification of Microarray Data

Milan, May 2003

Anestis Antoniadis

Laboratoire IMAG-LMC
University Joseph Fourier
Grenoble, France

Outline

- **Logistic regression**

Outline

- **Logistic regression**
- Classical

Outline

- **Logistic regression**
- Classical
- Multicollinearity

Outline

- **Logistic regression**
- Classical
- Multicollinearity
- Overfitting

Outline

- **Logistic regression**
- Classical
- Multicollinearity
- Overfitting
- **Penalized Logistic Regression**

Outline

- **Logistic regression**
- Classical
- Multicollinearity
- Overfitting
- **Penalized Logistic Regression**
- Ridge

Outline

- **Logistic regression**
- Classical
- Multicollinearity
- Overfitting
- **Penalized Logistic Regression**
- Ridge
- Other penalization methods

Outline

- **Logistic regression**
- Classical
- Multicollinearity
- Overfitting
- **Penalized Logistic Regression**
- Ridge
- Other penalization methods
- **Classification**

Outline

- **Logistic regression**
- Classical
- Multicollinearity
- Overfitting
- **Penalized Logistic Regression**
- Ridge
- Other penalization methods
- **Classification**
- An example of application

Introduction

- Logistic regression provides a good method for classification by modeling the probability of membership of a class with transforms of linear combinations of explanatory variables.

Introduction

- Logistic regression provides a good method for classification by modeling the probability of membership of a class with transforms of linear combinations of explanatory variables.
- Classical logistic regression does not work for microarrays because there are far more variables than observations.

Introduction

- Logistic regression provides a good method for classification by modeling the probability of membership of a class with transforms of linear combinations of explanatory variables.
- Classical logistic regression does not work for microarrays because there are far more variables than observations.
- Particular problems are multicollinearity and overfitting

Introduction

- Logistic regression provides a good method for classification by modeling the probability of membership of a class with transforms of linear combinations of explanatory variables.
- Classical logistic regression does not work for microarrays because there are far more variables than observations.
- Particular problems are multicollinearity and overfitting
- A solution: use penalized logistic regression.

Two-class classification

Situation: A number of biological samples have been collected, preprocessed and hybridized to microarrays. Each sample can be in one of two classes. Suppose that we have n_1 microarrays taken from one of the classes and n_2 microarrays taken from the other (e.g., 1=ALL, 2=AML); $n_1 + n_2 = n$.

Question: Find a statistical procedure that uses the expression profiles measured on the arrays to compute the probability that a sample belongs to one of the two classes and use it to classify any new array that comes along.

Solution: Logistic regression?

Logistic regression

Let Y_j indicate the status of array j, $j = 1, \ldots, n$ and let x_{ij}, $i = 1, \ldots, m$ be the normalized and scaled expressions of the m genes on that array.

Imagine that one specific gene has been selected as a good candidate for discrimination between the two classes (say $y = 0$ and $y = 1$).

If X is the expression measured for this gene, let $p(x)$ be the probabilty that an array with measured expression $X = x$ represents a class of type $Y = 1$.

A simple regression model would be

$$p(x) = \alpha + \beta x$$

with α and β estimated with a standard linear regression procedure.

Not a good idea!

- $p(x)$ may be estimated with negative values

A simple regression model would be

$$p(x) = \alpha + \beta x$$

with α and β estimated with a standard linear regression procedure.

Not a good idea!

- $p(x)$ may be estimated with negative values
- $p(x)$ may be larger than 1

Solution

Transform $p(x)$ to $\eta(x)$:

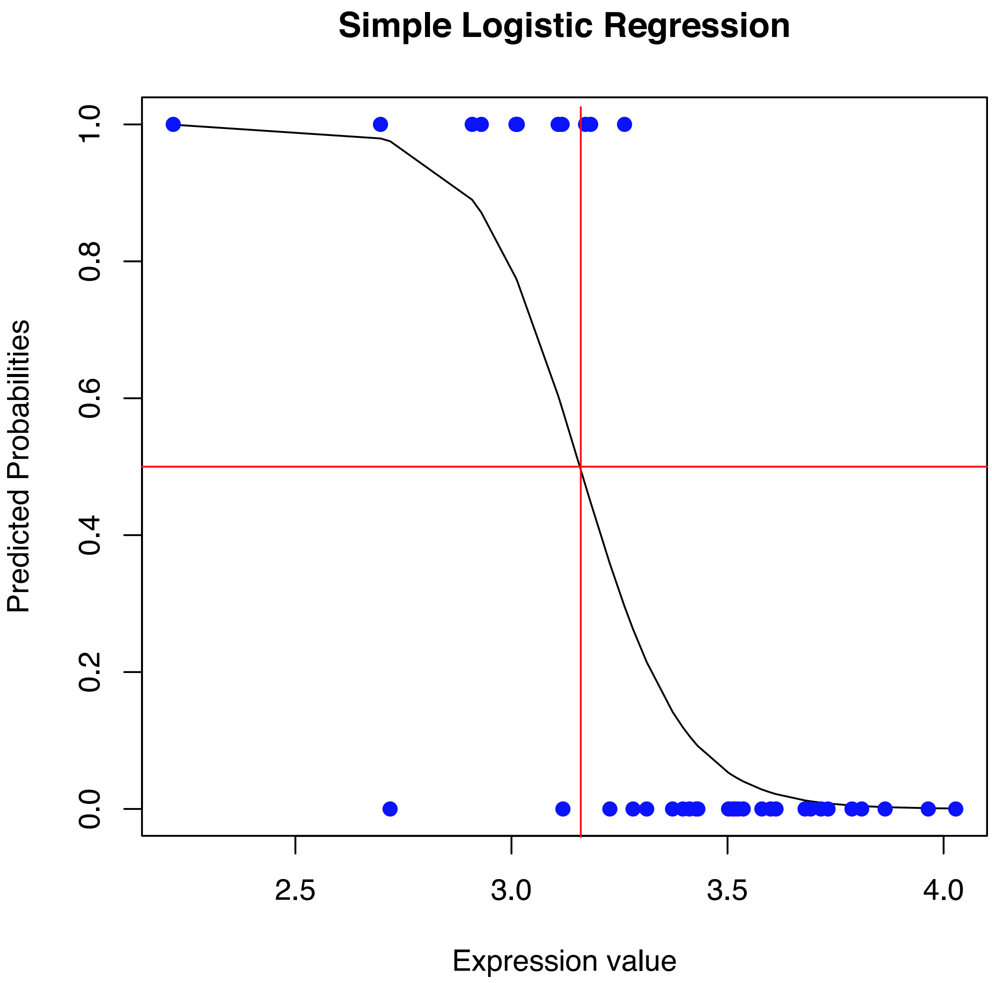
$$\eta(x) = \log \frac{p(x)}{1 - p(x)} = \alpha + \beta x.$$

The curve that computes $p(x)$ from $\eta(x)$,

$$p(x) = \frac{1}{1 + \exp(-\eta(x))}$$

is called the *logistic* curve.
This a special case of the generalized linear model. Fast and stable algorithms to estimate the parameters exist (`glm` package in R).

Example from the ALL/AML data



The estimated curve gives the probability of $y = 1$ (AML) for a given value of x. error rate = 13.16

It is straightforward to extend the model with more variables (genes expressions), introducing explanatory variables $x_1, x_2, \ldots, x_p$:

$$\eta(x_1, x_2, \ldots, x_p) = \log \frac{p(\mathbf{x})}{1 - p(\mathbf{x})} = \alpha + \sum_{i=1}^{p} \beta_i x_i$$

The maximum likelihood algorithm for glm's can be extended to this case very easily.

So why not use all expressions on an array to build a logistic regression model for class prediction?

Rule of thumb

A rule of thumb, for a moderate number of explanatory variables (less than 15), is that the number of observations n should at least be five times or more the number of explanatory variables. Here $n << m$! So there are many more unknowns than equations and infinitely many solutions exist.

Another problem is a perfect fit to the data (no bias but high variance). This is something to distrust because the large variability in the estimates produces a prediction formula for discrimination with almost no power!

Moreover the algorithm may be highly instable due to multicollinearities in the x's.

Is logistic regression doomed to failure?

Penalized logistic regression

Consider first a classical multiple linear regression model

$$\mathbf{Y} = \boldsymbol{\mu} + \boldsymbol{\epsilon} = X\boldsymbol{\beta} + \boldsymbol{\epsilon},$$

where $\mathbf{Y}$, $\boldsymbol{\mu}$ are $n \times 1$ and X is a $n \times m$ matrix. The n components of the mean $\boldsymbol{\mu}$ of $\mathbf{Y}$ are modeled as linear combinations of the m columns of X.
The regression coefficients are estimated by minimizing

$$S = \frac{1}{n} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{m} x_{ij}\beta_j)^2,$$

leading to

$$\hat{\boldsymbol{\beta}} = (X'X)^{-1} X'\mathbf{Y}$$

A large m may lead to serious overfitting.

A remedy

The key idea in *penalization* methods is that overfitting is avoided by imposing a penalty on large fluctuations on the estimated parameters and thus on the fitted curve.
Denote

$$S_n(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \sum_{j=1}^{m} x_{ij}\beta_j)^2 + \lambda J(\boldsymbol{\beta}),$$

where $J(\boldsymbol{\beta})$ is penalty that discourages high values of the elements of $\boldsymbol{\beta}$.

When $J(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_2^2 = \sum_{j=1}^{m} \beta_j^2$ the method is called quadratic regularization or ridge regression, and leads, for $\lambda > 0$, to the unique solution $\hat{\boldsymbol{\beta}}_\lambda = (X'X + \lambda I_m)^{-1}X'\mathbf{Y}$.

Penalizing logistic regression

$$\eta_i = \frac{p_i}{1 - p_i} = \alpha + \sum_{j=1}^{m} x_{ij}\beta_j$$

In the GLM jargon:

- η is the linear predictor.

Penalizing logistic regression

$$\eta_i = \frac{p_i}{1 - p_i} = \alpha + \sum_{j=1}^{m} x_{ij}\beta_j$$

In the GLM jargon:

- η is the linear predictor.
- the logistic function $\log(x/(1-x))$ is the canonical link function for the binomial family.

Penalizing logistic regression

$$\eta_i = \frac{p_i}{1-p_i} = \alpha + \sum_{j=1}^{m} x_{ij}\beta_j$$

In the GLM jargon:

- η is the linear predictor.
- the logistic function $\log(x/(1-x))$ is the canonical link function for the binomial family.

$\ell(\mathbf{y}, \boldsymbol{\beta}) = \sum_{i=1}^{n} y_i \log p_i + \sum_{i=1}^{n} (1-y_i)\log(1-p_i)$ is the log-likelihood and $-\ell + \frac{\lambda}{2} J(\boldsymbol{\beta})$ is the penalized negative log-likelihood

Denote by $\mathbf{u}$ the vector of ones in $\mathbb{R}^n$. By differentiation of the penalized -log-likelihood with respect to α and the β_j's follows

$$\begin{aligned} \mathbf{u}'(\mathbf{y} - \mathbf{p}) &= 0 \\ X'(\mathbf{y} - \mathbf{p}) &= \lambda \boldsymbol{\beta} \end{aligned}$$

The equations are nonlinear because of the nonlinear relation between p and α and $\boldsymbol{\beta}$. A first order Taylor expansion gives

$$p_i \sim \tilde{p}_i + \frac{\partial p_i}{\partial \alpha}(\alpha - \tilde{\alpha}) + \sum_{j=1}^{m} \frac{\partial p_i}{\partial \beta_j}(\beta_j - \tilde{\beta}_j)$$

Now

$$\begin{aligned} \frac{\partial p_i}{\partial \alpha} &= p_i(1 - p_i) \\ \frac{\partial p_i}{\partial \beta_j} &= p_i(1 - p_i)x_{ij} \end{aligned}$$

Using it and setting $W = \text{diag}(p_i(1 - p_i))$ we have

$$\begin{aligned} \mathbf{u}'\tilde{W}\mathbf{u}\alpha + \mathbf{u}'\tilde{W}X\boldsymbol{\beta} &= \mathbf{u}'(\mathbf{y} - \tilde{\mathbf{p}} - \tilde{W}\tilde{\boldsymbol{\eta}}), \\ X'\tilde{W}\mathbf{u}\alpha + (X'\tilde{W}X + \lambda I)\boldsymbol{\beta} &= X'(\mathbf{y} - \tilde{\mathbf{p}} - \tilde{W}\tilde{\boldsymbol{\eta}}) \end{aligned}$$

Suitable starting values are $\tilde{\alpha} = \log(\bar{y}/(1 - \bar{y}))$ and $\tilde{\boldsymbol{\beta}} = 0$.

Choosing λ

The choice of λ is crucial! Need a procedure that estimates the ”optimal” value of the smoothing or ridge parameter λ from the data.
Cross-validation: set apart some of the data, fit a model to the rest and see how well it predicts. Several schemes can be conceived. One is to set apart, say, one third of the data. More complicated is "leave-one-out" cross-validation: each of the n observations is set apart in turn and the model is fitted to the $m-1$ remaining ones. This is rather expensive as the amount of work is proportional to $m(m-1)$.

The performance of cross-validation one may use either the fraction of misclassification or the strength of log-likelihood prediction $\sum_i (y_i \log \hat{p}_{-i} + (1-y_i)\log(1-\hat{p}_{-i}))$.

AIC

Akaike's Information Criterion (AIC). Choose λ that balances rightly the complexity of the model and the fidelity to the data. AIC is defined as

$$AIC = \mathrm{Dev}(\mathbf{y}|\hat{\mathbf{p}}) + 2\mathrm{effdim},$$

where $\mathrm{Dev}(\cdot)$ is the deviance (equal to -2ℓ) and effdim is the effective dimension of the model. Hastie and Tibshirani estimate this by

$$\mathrm{effdim} = \mathrm{trace}(Z(Z'WZ + \lambda R)^{-1}WZ')$$

where $Z = [\mathbf{u}|X]$ and R is the $m+1 \times m+1$ identity matrix with $r_{11} = 0$.

Other choices of J

The behavior of the resulting estimate not only depends on λ but also on the form of the penalty function $J(\boldsymbol{\beta})$. Another form that one could consider is

$$J(\boldsymbol{\beta}) = \sum_k \gamma_k \psi(\beta_k) \text{ where } \gamma_k > 0.$$

Several penalty functions have been used in the literature.

Penalties

Typical examples are:

- The L_1 penalty $\psi(\beta) = |\beta|$ results in LASSO (first proposed by Donoho and Johnstone (1994) in the wavelet setting and extended by Tibshirani (1996) for general least squares settings).

Penalties

Typical examples are:

- The L_1 penalty $\psi(\beta) = |\beta|$ results in LASSO (first proposed by Donoho and Johnstone (1994) in the wavelet setting and extended by Tibshirani (1996) for general least squares settings).
- More generally, the L_q $(0 \leq q \leq 1)$ $\psi(\beta) = |\beta|^q$ leads to bridge regression (see Frank and Friedman (1993)).

Penalties

Typical examples are:

- The L_1 penalty $\psi(\beta) = |\beta|$ results in LASSO (first proposed by Donoho and Johnstone (1994) in the wavelet setting and extended by Tibshirani (1996) for general least squares settings).
- More generally, the L_q $(0 \leq q \leq 1)$ $\psi(\beta) = |\beta|^q$ leads to bridge regression (see Frank and Friedman (1993)).

Such penalties have the feature that many of the components of $\boldsymbol{\beta}$ are shrunk all the way to 0. In effect, these coefficients are deleted. Therefore, such procedures perform a model selection.

Conditions on ψ

Usually, the penalty function ψ is chosen to be symmetric and increasing on $[0, +\infty)$. Furthermore, ψ can be convex or non-convex, smooth or non-smooth.
In the wavelet setting, Antoniadis and Fan (2001) provide some insights into how to choose a penalty function. A good penalty function should result in

- *unbiasedness*,

Conditions on ψ

Usually, the penalty function ψ is chosen to be symmetric and increasing on $[0, +\infty)$. Furthermore, ψ can be convex or non-convex, smooth or non-smooth.
In the wavelet setting, Antoniadis and Fan (2001) provide some insights into how to choose a penalty function. A good penalty function should result in

- *unbiasedness*,
- *sparsity*

Conditions on ψ

Usually, the penalty function ψ is chosen to be symmetric and increasing on $[0, +\infty)$. Furthermore, ψ can be convex or non-convex, smooth or non-smooth.
In the wavelet setting, Antoniadis and Fan (2001) provide some insights into how to choose a penalty function. A good penalty function should result in

- *unbiasedness*,
- *sparsity*
- *stability*.

Examples

Penalty function	Convexity	Smoothness at 0	Authors
$\psi(\beta) = \|\beta\|$	yes	$\psi'(0^+) = 1$	(Rudin 1992)
$\psi(\beta) = \|\beta\|^\alpha, \;\; \alpha \in (0,1)$	no	$\psi'(0^+) = \infty$	(Saquib 1998)
$\psi(\beta) = \alpha\|\beta\|/(1+\alpha\|\beta\|)$	no	$\psi'(0^+) = \alpha$	(Geman 92, 95)
$\psi(0) = 0, \; \psi(\beta) = 1, \forall\beta \neq 0$	no	discontinuous	Leclerc 1989
$\psi(\beta) = \|\beta\|^\alpha, \; \alpha > 1$	yes	yes	Bouman 1993
$\psi(\beta) = \alpha\beta^2/(1+\alpha\beta^2)$	no	yes	McClure 1987
$\psi(\beta) = \min\{\alpha\beta^2, 1\}$	no	yes	Geman 1984
$\psi(\beta) = \sqrt{\alpha+\beta^2}$	yes	yes	Vogel 1987
$\psi(\beta) = \log(\cosh(\alpha\beta))$	yes	yes	Green 1990
$\psi(\beta) = \begin{cases} \beta^2/2 & \text{if} \quad \|\beta\| \leq \alpha, \\ \alpha\|\beta\| - \alpha^2/2 & \text{if} \quad \|\beta\| > \alpha. \end{cases}$	yes	yes	Huber 1990

Examples of penalty functions

An example

As in Eilers et al. (2002):

- The Golub data set preprocessed as in Dudoit et al. (2002)

An example

As in Eilers et al. (2002):

- The Golub data set preprocessed as in Dudoit et al. (2002)
- Prefilter the expressions in X with a floor of 100 and a ceiling of 1600.

An example

As in Eilers et al. (2002):

- The Golub data set preprocessed as in Dudoit et al. (2002)
- Prefilter the expressions in X with a floor of 100 and a ceiling of 1600.
- Eliminate columns of X with $\max/\min \leq 5$ and $\max - \min \leq 500$.

An example

As in Eilers et al. (2002):

- The Golub data set preprocessed as in Dudoit et al. (2002)
- Prefilter the expressions in X with a floor of 100 and a ceiling of 1600.
- Eliminate columns of X with $\max/\min \leq 5$ and $\max - \min \leq 500$.
- Take logarithms to base 10.

An example

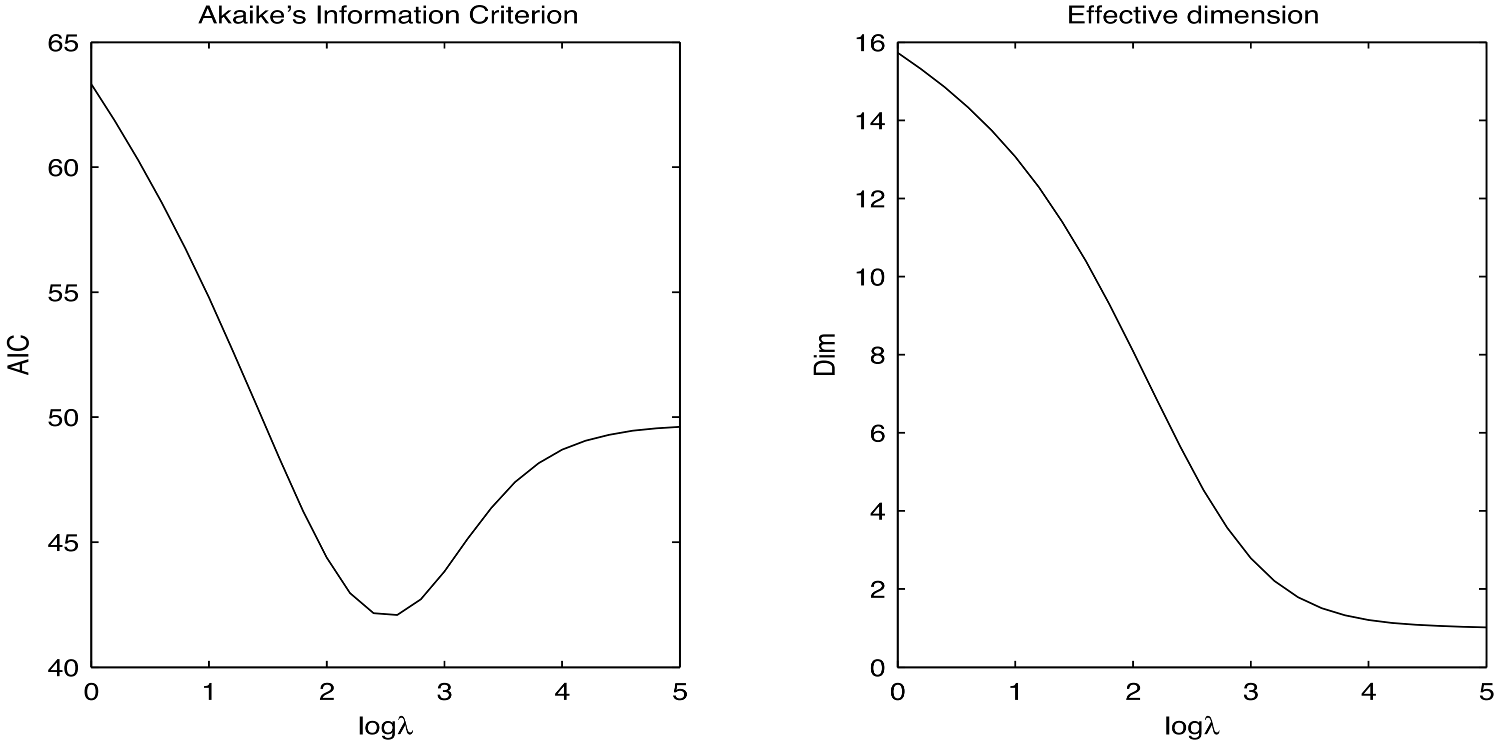
As in Eilers et al. (2002):

- The Golub data set preprocessed as in Dudoit et al. (2002)
- Prefilter the expressions in X with a floor of 100 and a ceiling of 1600.
- Eliminate columns of X with $\max/\min \leq 5$ and $\max - \min \leq 500$.
- Take logarithms to base 10.

We are left with $m = 3571$ genes and 38 arrays, 25 of them consisting of ALL and 11 of AML.

To find λ AIC was used. The range of λ was considered on a logarithmic scale grid from 0 to 5. The optimal value was $\lambda = 400$ and effdim corresponding to 4.5.

The choice of λ

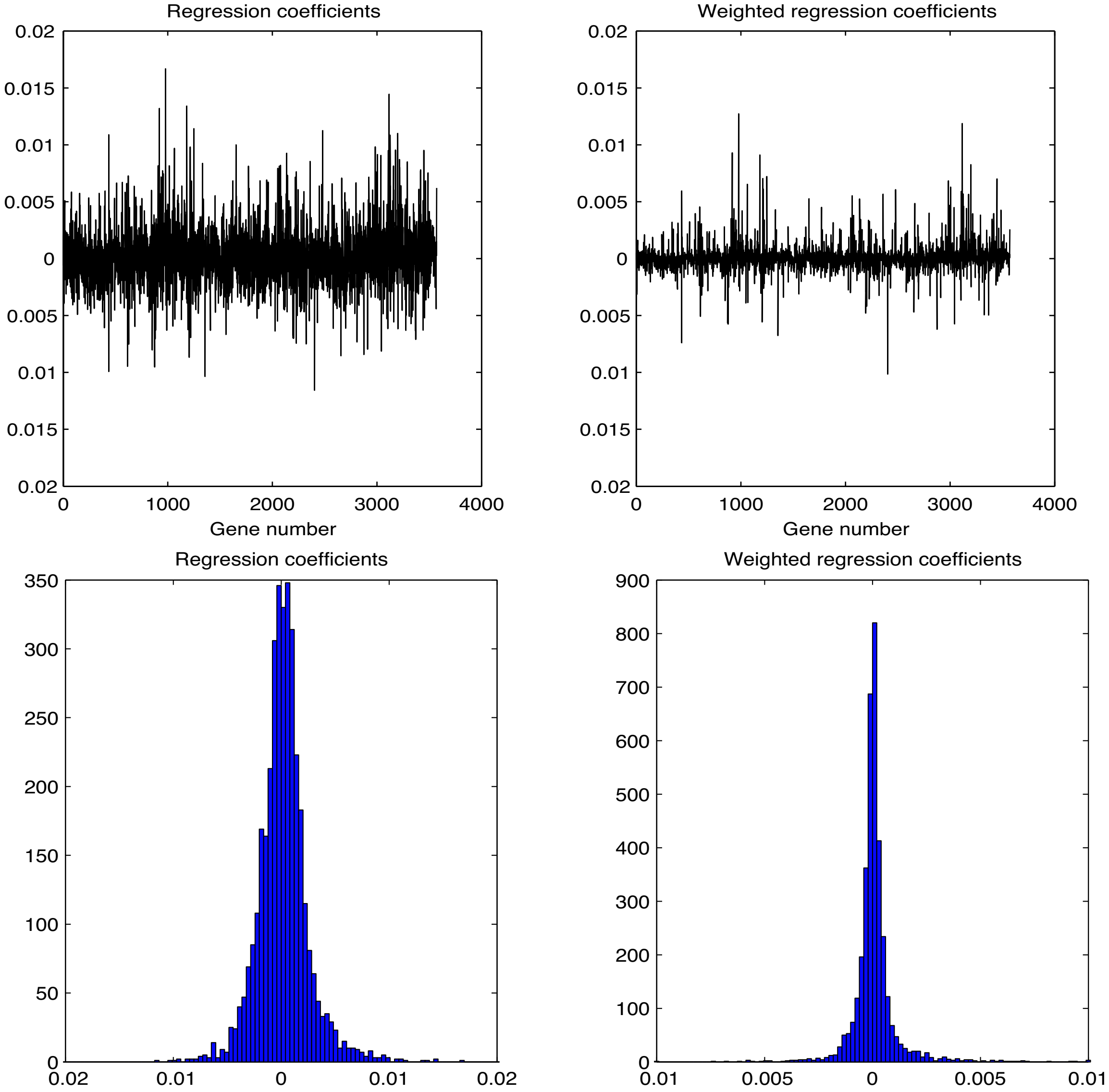


AIC and effdim as a function of $\log \lambda$

One can get an impression of the estimated coefficients by doing a sequence plot of them.

In order to get the same weight for the coefficients one also may display them by multiplying them by the standard deviation of the corresponding column of X in order to produce a scale invariant plot.

Estimated coefficients



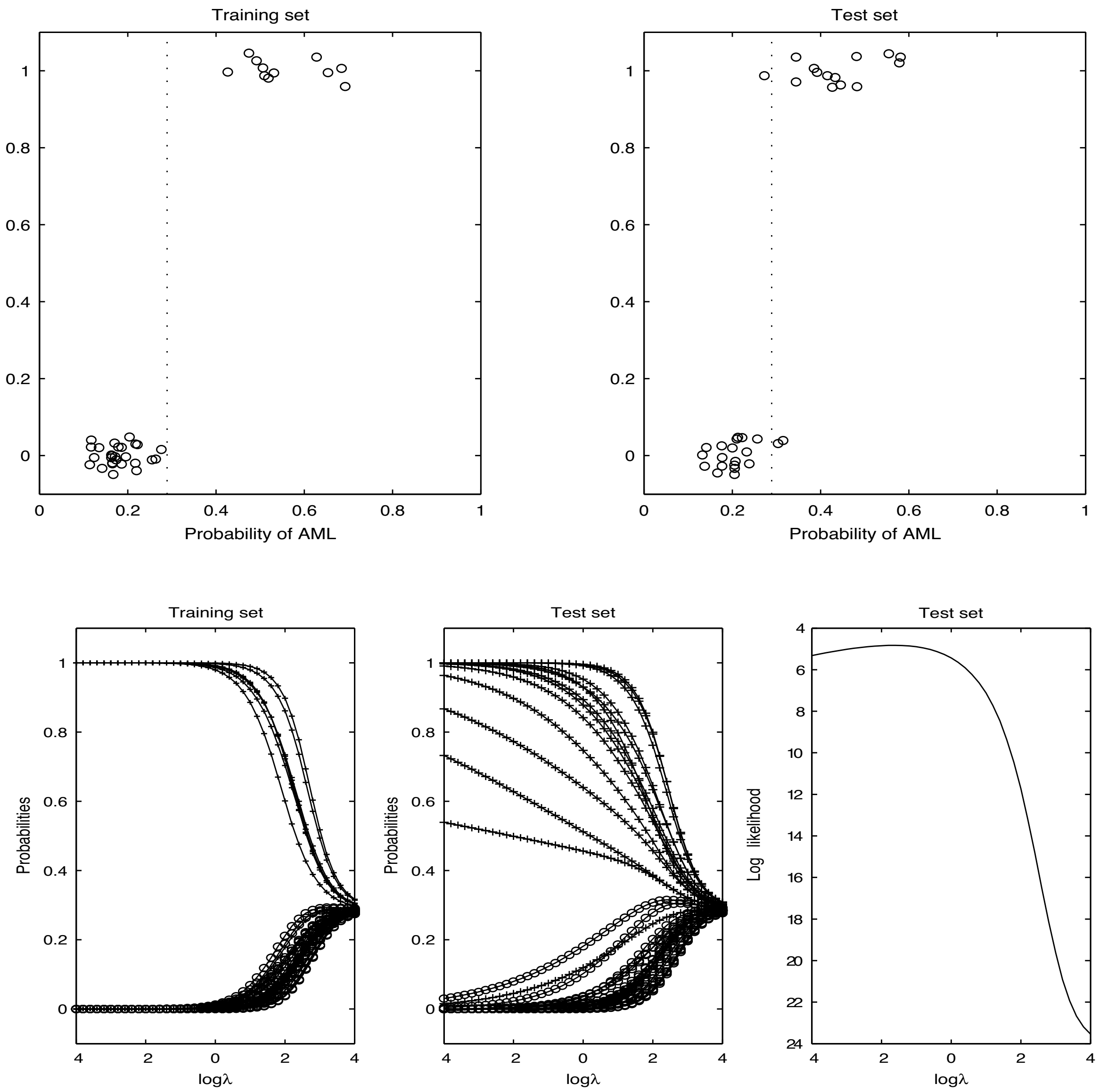
Estimated regression coefficients and their histograms

Without any knowledge of the expressions, but knowing that a fraction $\bar{y}$ of the arrays is from AML cases, one would classify an array as AML with probability $\bar{y}$.

Thus a very simple decision rule would be to compute $\hat{p}$ and see if it is lower of higher than $\bar{y}$.

The real test is the classification of new data, that were not used for the estimation of the model. The test set consist of 20 cases of ALL and 14 of AML. This rule would put three arrays in the wrong class.

Classification



Classification prob. as a function of λ. A vertical line at $p = \bar{y}$.

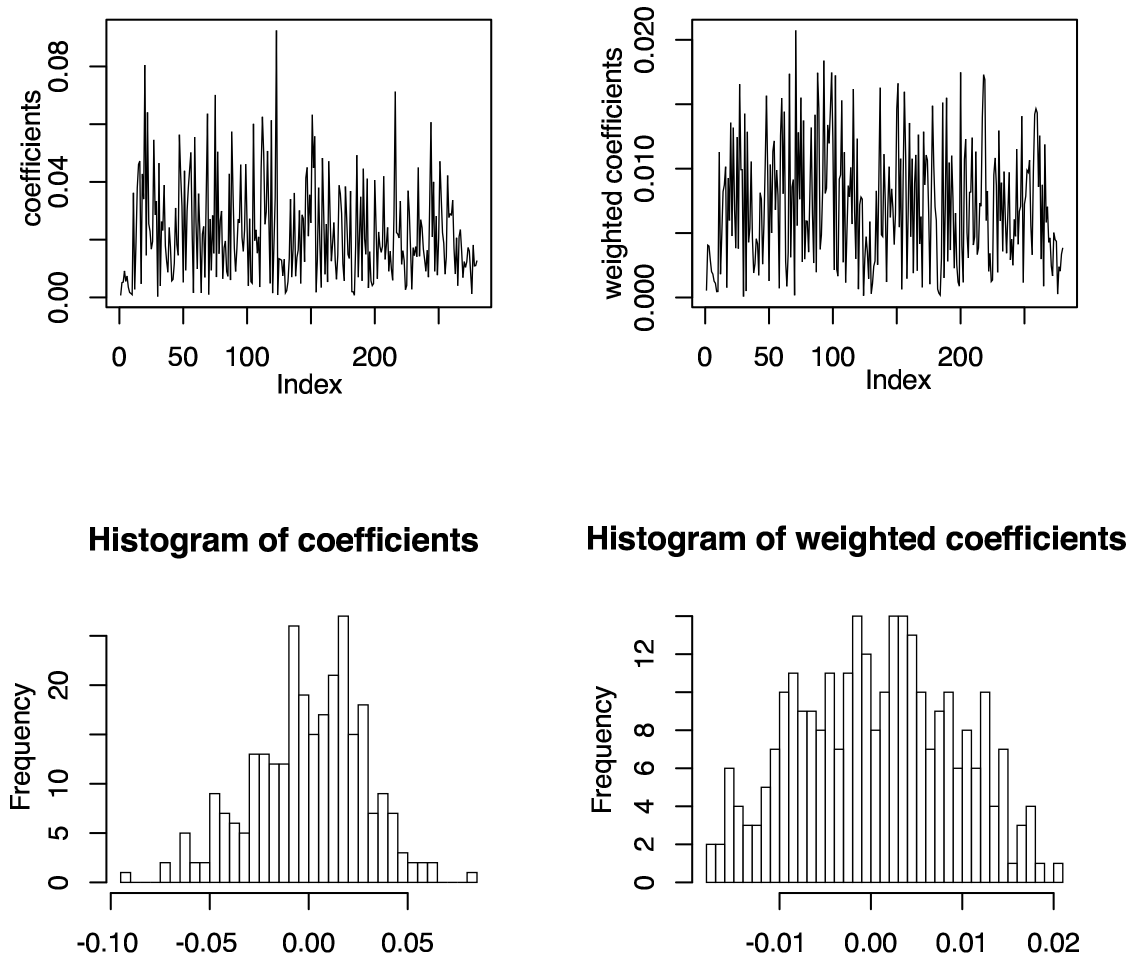
Same example with prefiltering

This time we have used Dudoit et al. (2002) procedures.

We pre-filter and select the 280 most discriminant genes (on the learning set).

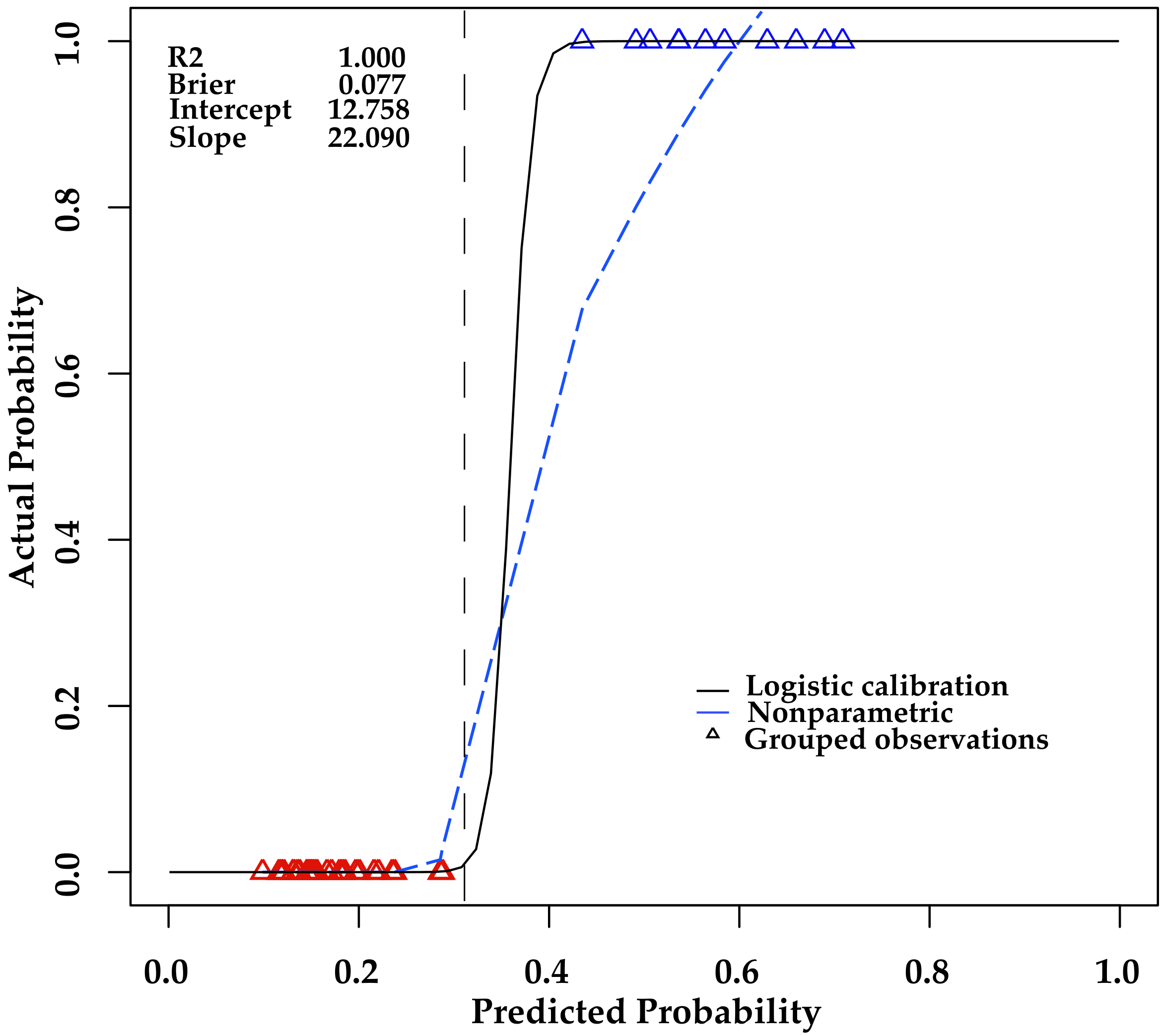
To perform the computations we have used the `lrm.fit` function in the Design package.

Estimated coefficients (280)



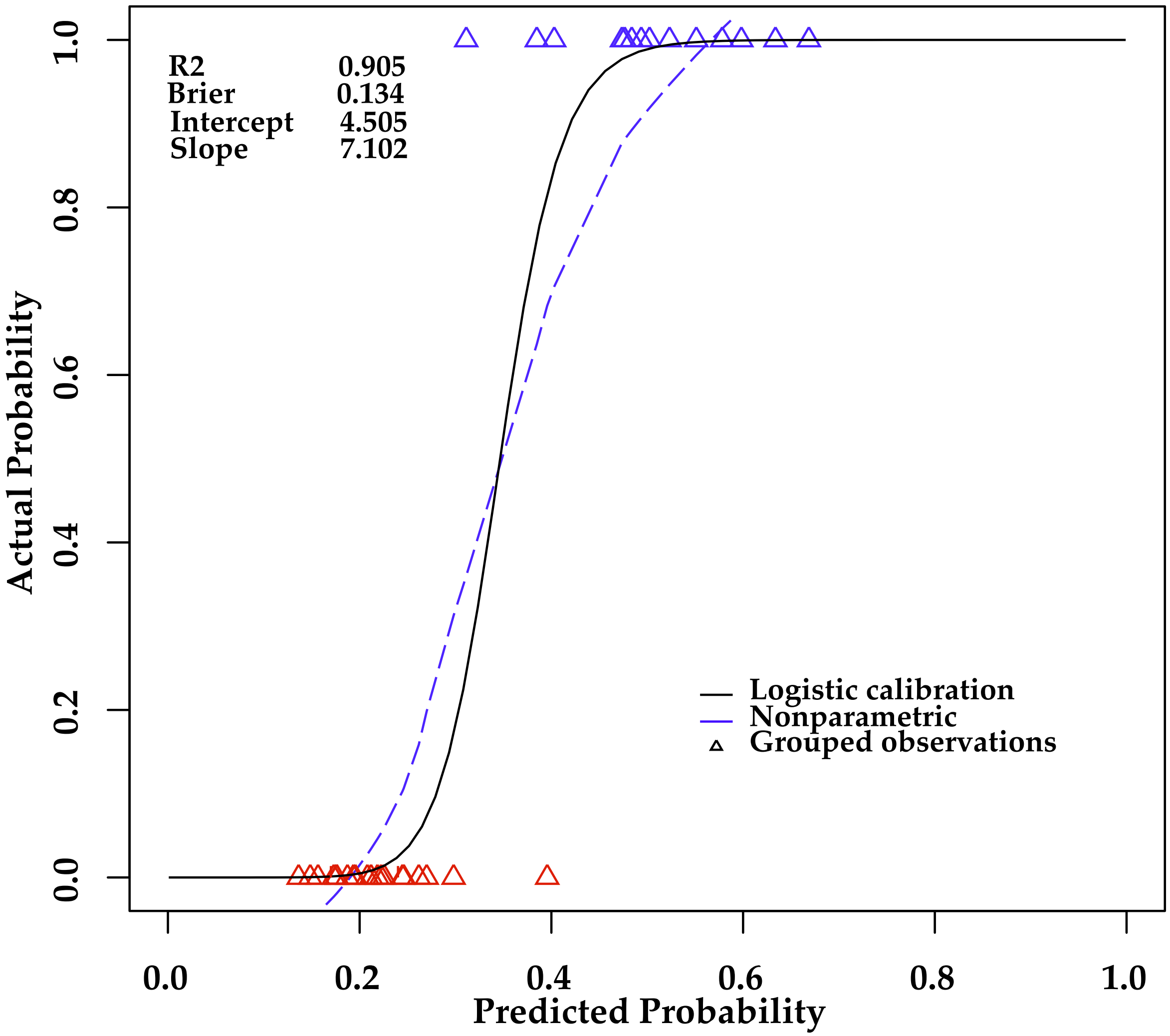
Estimated regression coefficients and their histograms

Classification (LS)



Predicted prob ($\lambda = 200$)

Classification (TS)



Predicted prob ($\lambda = 200$)

Some references

A. Antoniadis and J. Fan, JASA, 2001.

P. H. C. Eilers, J. M. Boerb, G-J. van Ommenb and H. C. van Houwelingen (2002). preprint.

T. R. Golub, D. K. Slonim, and P. T. et al., Science 286, pp. 531– 537, 1999.

A. E. Hoerl, R. W. Kennard, and R. W. Hoerl, Applied Statistics 34, pp. 114–120, 1985.

B. D. Marx and P. H. C. Eilers, Technometrics 41, pp. 1–13, 1999.

P. McCullagh and J. A. Nelder, Generalized Linear Models, 2nd edition, Chapman and Hall, London, 1989.

I. E. Frank and J. H. Friedman, Technometrics 35, pp. 109–148, 1993.

J. C. van Houwelingen and S. le Cessie, Statistics in Medicine 9, pp. 1303–1325, 1990.

K. P. Burnham and D. R. Anderson, Model Selection and Inference, Springer, New-York, 1998.

T. J. Hastie and R. J. Tibshirani, Generalized Additive Models, Chapman and Hall, London, 1990.

Tibshirani, R, Journal of the Royal Statistical Society, series B 58, 267-288, 1996.