Sparse multi-class prediction based on the Group Lasso in multinomial logistic regression

Thesis committee:

Dr. Drs. Jelle Goeman

Prof. Dr. Ir. Marcel Reinders

Dr. Drs. Erik van Zwet

Dr. Drs. Lodwijk Wessels

Drs. Wouter Meuleman

Written by:
Mathijs Sanders
Bioinformatics
Technical University of Delft
Delft, October 2009

Preface

Many classification procedures are based on variable selection methodologies. This master thesis concentrates on continuous variable selection procedures based on the shrinkage principle. Generally, we would like to find sparse prediction rules for multi-class classification problems such that in increases the prediction accuracy but also the interpretability of the obtained prediction rules. For these reasons we have chosen for the multinomial logistic regression model as its penalization procedures perform continuous variable selection and generally lead to sparse prediction rules. Next to the multinomial logistic regression model we have also implemented the Ridge penalization, Lasso penalization, Elastic net penalization, and the Group Lasso. The emphasis of this research lies on the Lasso and Group Lasso. The Lasso performs in a multi-class classification problem a variable selection on individual regression coefficients. In the multinomial regression model each predictor has a regression coefficient per class. The selection of the individual regression coefficients is less logical than the selection of an entire predictor. For this reason it could select redundant predictors leading to more retained predictors and less interpretable prediction rules. To overcome this problem we have developed a Group Lasso procedure with a novel group structure. The advantage of using the Group Lasso is that it performs variable selection on the predefined groups. In our model we developed a group structure which groups all the regression coefficients, i.e. of each class, of each predictor. This group structure facilitates the selection of an entire predictor. We demonstrate on the basis of gene expression profiles of 531 well-characterized Acute Myeloid Leukemia patients that the Group Lasso obtains less predictors with a similar prediction accuracy when compared to the regular Lasso. Finally, the Group Lasso facilitates the comparison of regression coefficients between classes for each predictor, which is not possible with the Lasso.

Acknowledgement

I would express my appreciation and gratitude to Jelle Goeman, who has been my mentor and supervisor during this study. I would like to thank him for giving me the opportunity to work at such a gentile and successful department. He has the uncanny ability to remain critical and explain difficult problems in way that I could understand them. I look forward to the many projects we could work on together. Next, I would like to express my gratitude to Marcel Reinders, my supervisor during the Master course. Marcel has the great ability to translate problems and descriptions, stemming not from his field of expertise, to terms that are understandable for him. For him this also works vice versa and during the master thesis he showed the great ability to highlight different aspects of the research of which we hadn't thought before. Furthermore I would like to thank my colleges from the University of Leiden for their support: Livio, Christian, Alina, Saskia, Lies, Erik, Theo, Gerard and all of those that I might have forgotten. I would also like to thank my peers and mentors from the Technical University of Delft: Rosa, Dick, Patrick, Tisha, Bas and Alex. Next, I would like to thank my colleagues from the Erasmus Medical Centre: Peter, Justine and Erdogan. Finally, I would like to thank Lodewijk Wessels, Wouter Meuleman and Erik van Zwet for spending their valuable time for reading this master thesis and being members of my Thesis committee.

Mathijs Sanders
Delft, October 2009

Table of Contents

Paper

Sparse multi-class prediction based on the Group Lasso in multinomial logistic regression

Mathijs Sanders

and

Jelle Goeman

Department of Medical Statistics and Bioinformatics, Leiden University Medical Center, Postzone S5-P, P.O. Box 9604, 2300 RC Leiden, The Netherlands

Summary: Continuous variable selection using shrinkage procedures have recently been considered as favorable models in a wide range of scientific research; in particular biomedical research. In some cases, it is desirable to select as few predictors as possible, to increase the interpretability of the attained prediction rule. One frequently used shrinkage procedure; the Lasso, imposes a L_1 regularization on the regression coefficients of general linear models, inherently leading to sparse prediction rules. When dealing with multi-class prediction in generalized linear models each predictor has a regression coefficient for each class. A major disadvantage is that the Lasso selects individual regression coefficients instead of the more logical selection of predictors. In this paper, we demonstrate a new regularization procedure, based on the Group Lasso in multinomial logistic regression. This results in a lower number of retained predictors, but with similar prediction accuracy when compared to the regular Lasso regularization. To illustrate the new regularization applicability we have employed it on a large cohort of acute myeloid leukemia patients (AML, n=531) who are characterized on a gene expression microarray.

1. Introduction

The emphasis in regression models is on finding explanatory variables, also called `predictors`, which can predict response variables accurately. Contemporary high-throughput technologies, have given rise to vast amounts of high-dimensional data. Given the high-dimensionality of the data, it is worthwhile to perform variable selection, as it would result in sparser prediction rules which could also be used for subsequent analysis. Best-subset procedures are in most cases computationally intensive; even for a moderate number of variables, and are known to be unstable due to their discrete nature. More robust strategies have been proposed for the multinomial logistic regression model (Krishnapuram *et al*., 2005) by imposing a penalty on the regression

coefficients; see Le Cessie (1990). In these models, based on the logistic regression model, each class has its own set of regression coefficients. By imposing penalizations, also called `regularization`, on these regression coefficients one can automatically control the behavior of these coefficients as the model is being fit. A frequently used penalization methodology is the Lasso (Tibshirani R., 1996); which puts a L_1 regularization on the regression coefficients. This regularization retains one predictor from a set of pair wise correlated predictors and discards the remaining. In the usual logistic regression set-up we have a continuous response $\boldsymbol{Y} \in \mathbb{R}^n$, an $n \times p$ design matrix X and a parameter vector $\boldsymbol{\beta} \in \mathbb{R}^{\boldsymbol{p}}$. This implies that we have p predictors and n observations. The Lasso estimator is then defined as:

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}} \left\{ -\ell(\boldsymbol{\beta}) + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \quad (1)$$

For large values of λ, some coefficients of $\widehat{\boldsymbol{\beta}}$ are put exactly to zero and are considered to be of no impact on the response variable. The sparse prediction rules obtained with the Lasso procedure is one of the reasons why it is frequently used for high-dimensional data (Zhu J. and Hastie T., 2004). In a multi-class classification problem, we have a regression coefficient vector for each class. A specific predictor is retained if one of its regression coefficients over all classes is unequal to zero. Table 1 illustrates an excerpt of the regression coefficients vectors for a four-class classification problem (chromosomal aberrations, classification case 1, Results). It shows the major disadvantage of Lasso regulation, as it selects individual regression coefficients instead of the more natural selection of predictors. This does not only result in less interpretable prediction rules, but also increases the number of selected predictors.

	Other	T(15;17)	T(8;21)	INV(16)	Gene Symbol
1553588_at	9.55E-05	0	0	-0.0003	ND3
200026_at	9.91E-05	0	0	0	RPL34
200665_s_at	0	0	0	0.000659	SPARC
201324_at	-0.00024	0	0	0	EMP1
201360_at	-0.00014	0	0	0.000246	CST3
201432_at	0.00173	0	0	-0.00039	CAT
201502_s_at	0.000318	0	0	0	NFKBIA
201721_s_at	0	0	-0.00053	0	LAPTM5
202746_at	0	0	0	0.000388	ITM2A
202859_x_at	0	0	0.000122	0	IL8
202902_s_at	0	0	0	0.000201	CTSS
202917_s_at	0	0	0	0.00021	S100A8
203535_at	0	0	0	0.000762	S100A9

Table 1 Regression coefficients of a 4-class classification problem: Lasso has a tendency to set many regression coefficients to zero.

An alternative model the Group Lasso (Yuan and Lin, 2006; Meier, 2008) can overcome this problem by defining a suitable penalization function. This penalization procedure has been observed as an intermediate between Lasso and Ridge regulation and in addition has the attractive property to perform variable selection on predefined groups of predictors. Most logistic regression models, which had hitherto solely been based on single predictors, can now be replaced by entities reflecting grouping structures. This predefined grouping has given the possibility to integrate prior knowledge into the model and create structures relevant to research; such as pathway analysis. The elastic net (Zhou, 2005) was developed to take advantage of the grouping effect; however it lacks the ability to predefine group structures, which could inherently increase the interpretability of the prediction rule.

In this paper we have extended the Group Lasso for the logistic regression model to multi-class classification. In addition, we impose a group structure such that an entire predictor is retained or discarded over all classes. This implies that retained predictors have a regression coefficient unequal to zero for all classes and when discarded are all simultaneously set to zero. A benefit of this is that the coefficients can now be compared between classes of one particular predictor. The aim of the current study is to demonstrate that the new regularization procedure has a prediction accuracy comparable to that of the regular Lasso penalization, and in addition, that the optimal estimator contains less predictors. To demonstrate this we make use of the gene expression data from a large cohort of AML patients (n=531), with a distinct molecular-specific subtypes which can be used as classification objectives. The core of this algorithm will be explained further in Section 2. Next to the derivation of the algorithm we also address parameter identifiability problems and an approach for solving these issues. The reparameterization of the model has lead to the decision for a Quasi-Newton optimizer with bounding box constraints for the optimization. Section 3 presents the results with additional interpretation of the prediction rules. Finally, Section 4 discusses the extensions of the algorithm and concluding remarks.

2. Multinomial Group Lasso

2.1 Multinomial logistic regression

Firstly, a brief summary of the multinomial regression model is needed to fully understand the Group Lasso. The multinomial regression model is a multi-class classification procedure, which predicts the probability of a class by fitting the data to a logistic curve. Initially, we have a specific number of observations; n, a specific number of predictors belonging to these observations; p, and each observation can be assigned to g outcome categories. In an example, we have

the outcomes $Y_1, \cdots, Y_n$ for each observation and a corresponding $n \times p$ design matrix of predictors X. It is convenient to rewrite the outcomes to indicator functions which correspond to class participation. We define $y_{is} = \mathbf{1}_{\{Y_i = s\}}$ $(i = 1, \cdots, n; s = 1, \cdots, g)$, noting that each class has its own regression coefficients vector, $\boldsymbol{\beta_i} \in \mathbb{R}^p$ $(i = 1, \cdots, g)$**.** The corresponding probability is given by:

$$P(Y_i = s) = \mu_{is} = \frac{e^{\boldsymbol{\beta_{0s}} + \boldsymbol{x_i'\beta_s}}}{\sum_{t=1}^{\boldsymbol{g}} e^{\boldsymbol{\beta_{0t}} + \boldsymbol{x_i'\beta_t}}} \quad (2)$$

The model defined in (2) is overparameterized. Replacing $(\beta_{k1}, \cdots, \beta_{kg})$ by $(\beta_{k1} + c, \cdots, \beta_{kg} + c)$, for any $k \in \{1, \cdots, p\}$ and $c \in \mathbb{R}$, results in the same probabilities. Commonly, this problem is solved by defining one outcome category as the reference category, by setting all its coefficients to zero, i.e. $\beta_{1i} = \cdots = \beta_{pi} = 0$, for any $i \in \{1, \cdots, g\}$. The choice of reference category facilitates that the interpretation of the resulting parameter estimates. Instead of choosing a reference category, we will treat the outcome categories as symmetrical (Goeman *et al.*, 2006), as penalized models are not invariant to setting reference categories and inherently results in different prediction rules. Furthermore, the penalized general linear models are not affected by overparameterization in terms of function identifiability problems. For notational convenience we rewrite the regression coefficient vectors into a long vector format: $\boldsymbol{\beta}^* = (\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta_g})'$. We also rewrite y_{is}, μ_{is} into $ng \times 1$ vectors: $\boldsymbol{y} = (y_{11}, \cdots, y_{n1}, \cdots, y_{1g}, \cdots, y_{ng})$, $\boldsymbol{\mu} = (\mu_{11}, \cdots, \mu_{n1}, \cdots, \mu_{1g}, \cdots, \mu_{ng})$ and the design matrix into $\boldsymbol{X} = X \otimes I_g$, where $\otimes$ is the Kronecker product. The log-likelihood of this model is:

$$\ell(\boldsymbol{\beta}^*) = \sum_{i=1}^{n} \sum_{s=1}^{g} y_{is}\, log(\mu_{is}) \quad (3)$$

which has the gradient $\frac{\partial l(\beta)}{\partial \beta} = \boldsymbol{X'}(\boldsymbol{y} - \boldsymbol{\mu})$ and the Hessian $\frac{\partial^2 l(\beta)}{\partial \beta \partial \beta^T} = -\boldsymbol{X'WX}$. The $ng \times ng$ matrix W is given by:

$$\boldsymbol{W} = \begin{bmatrix} W^{11} & W^{12} & \cdots & W^{1g} \\ W^{21} & W^{22} & & \vdots \\ \vdots & & \ddots & \\ W^{g1} & \cdots & & W^{gg} \end{bmatrix}$$

Where

$$\operatorname{diag}(W^{st}) = \operatorname{diag}(W^{ts}) = \begin{cases} (-\mu_{1s}\mu_{1t}, \cdots, -\mu_{ns}\mu_{nt})' & \text{if } s \neq t \\ \left(\mu_{1s}(1-\mu_{1s}), \cdots, \mu_{ns}(1-\mu_{ns})\right)' & \text{if } s = t \end{cases}$$

Due to the convexity of the problem, we can use a Newton-Raphson algorithm to maximize the likelihood. A problem due to the overparameterization of the model, is that the Hessian is singular. By using Moore-Penrose or projection procedures, we can resolve this issue, but this is not a concern for the Group Lasso as earlier stated: penalized models are not affected by overparameterization in terms of function identifiability problems.

2.2 Penalty structure

The penalized log-likelihood under Lasso regulation (1), imposes a L_1 regularization on each individual regression coefficient per predictor over all classes in a multi-class prediction problem. A large amount of these regression coefficients are set to zero under strong penalization, resulting in sparse prediction rules. Its emphasis is on the selection of individual regression coefficients, instead of the selection of single predictors, leads to a larger number of retained predictors than desired. In addition, most regression coefficients per predictor are set to zero; this prohibits the comparison of the impact of the predictor on all predefined classes. To resolve this disadvantage we propose a new penalization structure based on the Group Lasso in multinomial logistic regression (Yuan and Lin, 2006; Meier L. *et al.*, 2008). The Group Lasso regularization structures allow the definition of groups as entities of the model. Instead of selecting single predictors, the model now selects predefined groups, facilitating different interpretations of the prediction rules. It does so by defining group structures based on the regression coefficients. Let us first define the beta matrix, of which the columns consist of regression coefficient vectors for each class:

$$\tilde{\beta} = \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1g} \\ \beta_{21} & \beta_{22} & & \beta_{2g} \\ \vdots & & \ddots & \vdots \\ \beta_{p1} & \beta_{p2} & \cdots & \beta_{pg} \end{bmatrix} = [\boldsymbol{\beta}_1 \quad \boldsymbol{\beta}_2 \quad \cdots \quad \boldsymbol{\beta}_g]$$

This beta matrix gives the opportunity to define many group structures, and was the underlying mechanism for the development of the Group Lasso. In this study, we would like to retain or discard each predictor; i.e. each row of this matrix, by setting all regression coefficients simultaneously unequal or equal to zero. This is accomplished by defining each row vector of regression coefficients as a group. Let us assume that we have a p-dimensional feature vector $\boldsymbol{x}_i \in \mathbb{R}^{\boldsymbol{p}}$, which

consists out of J groups. Let us denote by df_j the degrees of freedom of group j, rewrite $\boldsymbol{x}_i = \left(\boldsymbol{x}'_{i,1}, \boldsymbol{x}'_{i,2}, \cdots, \boldsymbol{x}'_{i,J}\right)'$ and denote the group of variable $\boldsymbol{x}'_{i,j} \in \mathbb{R}^{\boldsymbol{df_j}}$, $j = 1, \cdots, J$. The regression coefficient vector is parameterized as $\boldsymbol{\beta}_t = \left(\beta_{0t}, \boldsymbol{\beta}_{1,t}, \boldsymbol{\beta_{2,t}}, \cdots, \boldsymbol{\beta}_{J,t}\right)'$, $t = 1, \cdots, G$.

Given these groups we rewrite (2) as:

$$P(Y_i = s) = \mu_{is} = \frac{e^{\beta_{0s} + \sum_{j=1}^{J} \boldsymbol{x}'_{i,j} \boldsymbol{\beta}_{j,s}}}{\sum_{t=1}^{\boldsymbol{g}} e^{\beta_{0t} + \sum_{j=1}^{J} \boldsymbol{x}'_{i,j} \boldsymbol{\beta}_{j,t}}} \quad (4)$$

The Group Lasso estimator β_λ is given by the maximizer of the function:

$$\ell_{glasso}(\boldsymbol{\beta}^*; \lambda) = \ell(\boldsymbol{\beta}^*) - \lambda \sum_{j=1}^{J} \left\| \boldsymbol{\beta}_j \right\|_2 = \ell(\boldsymbol{\beta}^*) - \psi(\boldsymbol{\beta}^*) \; (5)$$

Hence, the penalty function sums the norm of each row of the beta matrix $\tilde{\beta}$. Note that Meier L. (2008) as well as Yuan M. (2006) integrate the square root of the degrees of freedom of each group in the summation. Given the current group structure, each group has the same degrees of freedom, thus the additional term is omitted.

2.3 Group Lasso estimator

To optimize the penalized log-likelihood function (5), the low-memory BFGS algorithm (L-BFGS-B, Liu (1989)) is used. This particular algorithm is a Quasi-Newton algorithm, as it needs a limited number of previous function and gradient evaluations to estimate the inverse Hessian. The gradient of the penalized log-likelihood function is given by:

$$\frac{\partial \ell_{glasso}(\boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}^*} = \frac{\partial \ell(\boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}^*} - \lambda \frac{\partial \psi(\boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}^*} \; (6),$$

where the gradient of the penalty function is defined as:

$$\frac{\partial \psi(\boldsymbol{\beta}^*)}{\partial \beta_{ij}} = \frac{\partial}{\partial \beta_{ij}} \left(\sqrt{\beta_{11}^2 + \cdots + \beta_{1g}^2} + \cdots + \sqrt{\beta_{p1}^2 + \cdots + \beta_{pg}^2} \right) = \frac{\beta_{ij}}{\sqrt{\beta_{i1}^2 + \cdots + \beta_{ig}^2}} \; (7)$$

2.4 Reparameterization and parameter identifiability

Optimizing the penalized log-likelihood function leads to major problems, as the function is only strictly convex and continuous in the internal space of all subspaces of the regression coefficients. The derivative of the penalized log-likelihood function remains undefined when one of the regression coefficients equals zero. This is issue is resolved by reparameterizing the model to a higher dimension where the function is strictly convex and continuous. The following reparameterization is proposed:

$$\begin{gathered}\beta_{ij} = \beta_{ij}^{+} - \beta_{ij}^{-} \\ \beta_{ij}^{+} = \max\left(\beta_{ij}, 0\right) \\ \beta_{ij}^{-} = -\min\left(\beta_{ij}, 0\right) \\ \beta_{ij}^{+} \geq 0 \\ \beta_{ij}^{-} \geq 0\end{gathered}$$

The reparameterization is realized by decomposing the individual regression coefficients into a positive part function (PPF) and a negative part function (NPF). These functions are constrained by the fact that each must be non-negative. For this reason we make use of the box constraints that can be set for the L-BFGS-B algorithm. Note that at the convergence either the PPF, NPF or both should be equal to zero. This reparameterization results in a model with twice as many parameters, which are restricted to a subspace of non-negative regression coefficients. As stated, in this single subspace the penalized log-likelihood function is strictly convex, continuous, and is differentiable in each internal point. Hence, instead of dealing with distinct continuous subspaces where the function is non-differentiable at their borders, i.e. when one of the regression coefficients is set to zero, we now have one subspace where the function is differentiable in its internal space. The log-likelihood gradient remains unchanged under the reparameterization, but the penalty function gradients are given by:

$$\begin{gathered}\frac{\partial\psi(\boldsymbol{\beta}^{*})}{\partial\beta_{ij}^{+}} = \frac{\partial\psi(\boldsymbol{\beta}^{*})}{\partial\beta_{ij}}\frac{\partial\beta_{ij}}{\partial\beta_{ij}^{+}} = \frac{\partial\psi(\boldsymbol{\beta}^{*})}{\partial\beta_{ij}} \\ \frac{\partial\psi(\boldsymbol{\beta}^{*})}{\partial\beta_{ij}^{-}} = \frac{\partial\psi(\boldsymbol{\beta}^{*})}{\partial\beta_{ij}}\frac{\partial\beta_{ij}}{\partial\beta_{ij}^{-}} = -\frac{\partial\psi(\boldsymbol{\beta}^{*})}{\partial\beta_{ij}}\end{gathered}$$

A problem occurs when all the regression coefficients of a group become zero, as the penalty function is no longer differentiable. To solve this problem the following limit is taken for the sake of continuity:

$$\lim_{\beta_{ij}\to 0}\frac{\beta_{ij}}{\sqrt{\beta_{i1}^{2} + \cdots + \beta_{ig}^{2}}} = 1, \;\; if\; \beta_{i1} = \cdots = \beta_{i(j-1)} = \beta_{i(j+1)} = \cdots = \beta_{ig} = 0$$

Next to the reparameterization, the optimization of the penalized log-likelihood is also affected by a parameter identifiability problem. The penalty function $\psi(\boldsymbol{\beta}^*)$ (5-7) consists of the norms of the row vectors of the beta matrix $\tilde{\beta}$. These norms are characterized by the squared regression coefficients β_{ij}^2, belonging to their respective groups. Under the reparameterization this squared regression coefficient is given by:

$$\beta_{ij}^2 = \left(\beta_{ij}^+ - \beta_{ij}^-\right)^2 = \beta_{ij}^{+2} - 2\beta_{ij}^+\beta_{ij}^- + \beta_{ij}^{-2} \ (8),$$

In equation (8), multiple instances of β_{ij}^+ or β_{ij}^- could give the exact same β_{ij}^2. This problem can be resolved by imposing a constraint on this equation. At convergence either the PPF, NPF or both should be equal to zero. This implies that the middle term of the factorization of β_{ij}^2 should be forced to be zero. This leads to the redefinition of equation (8):

$$\beta_{ij}^2 = \beta_{ij}^{+2} + \beta_{ij}^{-2} \ (9)$$

As we are trying to redefine the penalty function it is more appropriate to rewrite the penalized log-likelihood function:

$$\ell_{glasso}(\boldsymbol{\beta}^*; \lambda) = \ell(\boldsymbol{\beta}^*) - \lambda \sum_{j=1}^{J} \sqrt{\left\|\boldsymbol{\beta}_j^+\right\|_2^2 + \left\|\boldsymbol{\beta}_j^-\right\|_2^2} \ (10)$$

It is easily shown through the triangle-inequality that:

$$\sqrt{\left\|\boldsymbol{\beta}_j^+\right\|_2^2 + \left\|\boldsymbol{\beta}_j^-\right\|_2^2} \geq \sqrt{\left\|\boldsymbol{\beta}_j^+ - \boldsymbol{\beta}_j^-\right\|_2^2} = \sqrt{\left\|\boldsymbol{\beta}_j^+\right\|_2^2 - 2\left(\boldsymbol{\beta}_j^+\right)^T\boldsymbol{\beta}_j^- + \left\|\boldsymbol{\beta}_j^-\right\|_2^2} \ (11)$$

Hence, the redefinition of the penalty function $\psi(\boldsymbol{\beta}^*)$ is always larger or equal than its original definition. Given inequality (11) and the fact that either the PPF, NPF or both are zero at convergence, the redefined penalty function becomes equal to the original definition. By this redefinition we have solved the parameter identifiability problem and proven to be exactly the same as the original definition at convergence, we obtain the exact same prediction rules without convergence problems.

Table 2 illustrates an excerpt of the results from the same 4-class classification problem (chromosomal aberrations, classification case 1, Results) based on the modified Group Lasso. In comparison with Table 1 it immediately becomes clear that: (i) the number of predictors is decreased (ii) no regression coefficient of the retained predictors is set to zero, and (iii) the new group structure facilitates comparison of the regression coefficients between classes.

	Other	T(15;17)	T(8;21)	INV(16)	Gene Symbol
1553588_at	0.00018085	-8.59E-05	5.97E-05	-0.0001546	ND3
200665_s_at	-0.00014592	-1.73E-05	-7.23E-05	0.000235584	SPARC
201324_at	-0.00017254	1.18E-05	2.64E-05	0.000134331	EMP1
201360_at	-0.00020149	9.67E-06	-6.17E-05	0.000253532	CST3
201432_at	0.000946723	-0.00025838	-3.35E-05	-0.0006548	CAT
201502_s_at	0.000131047	-0.00012284	6.43E-05	-7.25E-05	NFKBIA
201721_s_at	0.000325746	1.74E-05	-0.00034902	5.93E-06	LAPTM5
202746_at	-0.00012466	1.67E-05	-0.00011551	0.000223436	ITM2A
202902_s_at	-7.06E-06	-1.00E-05	-5.58E-06	2.27E-05	CTSS
202917_s_at	-6.06E-05	-0.0001884	-8.16E-05	0.000330612	S100A8
203535_at	-0.00018007	-6.25E-05	-9.28E-05	0.000335433	S100A9

Table 2 Regression coefficients of a 4-class classification problem with the modified Group Lasso: The Group Lasso procedure produces sparser prediction rules. Furthermore it facilitates the comparison of regression coefficients between classes.

3. Results

AML is not a single disease, but a group of neoplasms with various genetic aberrations and variable prognosis and responses to treatment. The search for novel molecular markers is essential for therapeutical decision-making. A large number of molecular markers have been identified in the last decade, however the underlying mechanism of leukomogenesis still remains elusive. With the use gene expression profiling (GEP), the challenge lies in generating reliable prediction rules that can discriminate the different gene expression profiles and subsequently the variable subtypes of AML; for instance for the improvement of treatment decisions. We have applied our algorithm to the GEP of 540 clinically and molecularly well-characterized cases of AML. They originate from two different cohorts, the first of which represents a subset of 285 previously analyzed patients (Valk P. *et al.,* 2003), while the second was subsequently generated as a complement to the first. All samples are analyzed using the Affymetrix Human Genome U133APlus 2.0 GeneChips (Affymetrix, Santa Clara, CA, USA). All clinical, cytogenetic and molecular information as well as the gene expression data are readily available at the Gene Expression Omnibus (www.ncbi.nlm.nih.gov/geo, accession number GSE6981). All data has been preprocessed as described in Verhaak R. *et al.* (2009). AML cohort 1 (n=269) has been used as training set while AML cohort 2 (n=261) is subsequently used as test set. The optimal value for the regularization parameter λ was determined by 5-fold cross-validation. The gene expression signatures are available in Supplementary Tables S1-S2.

3.1 Classification case 1: Chromosomal aberrations and CEBPa

3.1.1 Background and Classification objective

The first classification objective is to classify AML patients with a favorable risk, i.e. with a translocation 15-17 (t(15;17)), translocation 8-21 (t(8;21)), inversion 16 (INV(16)). In addition to the classes with chromosomal aberrations, a class was created for samples that harbored a mutation in the well-known transcription factor CEBPa. This transcription factor is associated with inhibiting granulocyte differentiation. Usually, these classes are mutually exclusive, and no overlapping biological mechanisms influencing the gene expression patterns were expected. Finally, an additional class 'Other', was created for the remainder of the samples. Table 3 depicts the distribution of the different classes over the two cohorts.

Classes	AML cohort 1 (n=261)	AML cohort 2 (n=264)	Risk
Other	180 (70%)	204(77%)	Intermediate
t(15;17)	18(7%)	7(3%)	Favorable
t(8;21)	22(8%)	16(6%)	Favorable
inv(16)	23(8%)	18(7%)	Favorable
CEBPa	18(7%)	18(7%)	Favorable

Table 3: Distribution of the AML samples over the predefined classes.

3.1.2 Results

We have applied the global test for multinomial logistic regression (Goeman J.J., 2004) to investigate whether the fit of the model can discriminate the classes based on the given predictors, i.e. genes. This test can determine whether the global expression pattern of all genes is significantly related to the outcomes, i.e. class labels. It can be shown that the given genes are significantly related to the outcomes ($p < 0.0001$), hence giving positive evidence that the classes can be discriminated from each other. Using 5-fold cross-validation the optimal regularization parameter λ was set to 50. This resulted in 74 retained probe sets (Supplementary S1). Figure 1 illustrates the estimated test error curve for a grid for eleven evaluations of λ. The optimal regularization parameter λ for the Lasso penalization was determined by the same cross-validation procedure. The regularization parameter was set at 0.02 with 75 retained probe sets (Supplementary S3). For this classification case it does not matter whether to select the Lasso or the modified Group Lasso, when only depending on the number of retained predictors. The retained predictors of both procedures greatly overlap, with the exception of a few predictors. Strikingly, the Lasso makes four additional miss-classifications compared to the Group Lasso (Table 5 vs. Supplementary S6).

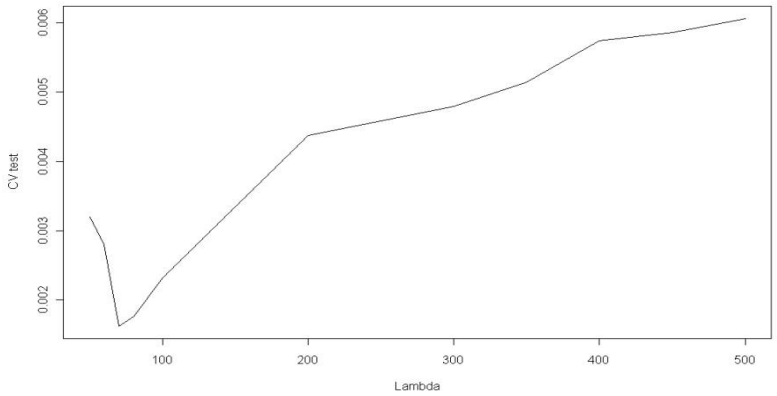


Figure 1. Estimated test error curve based on 5-fold cross-validation

All AML subtypes harboring chromosomal aberrations (t(15;17), t(8;21) and INV(16)) were predicted with 100% accuracy (Table 5), which was consistent with previous work. A substantial proportion of the samples with a CEBPa mutation were classified as being in the 'Other' category. After further investigation it became apparent that the misclassified samples all contain a single mutation instead of the more common double mutation; affecting both alleles. In previous work (Wouters B. *et al*. 2009) it was noted that double, but not single mutated samples have a distinct GEP and can be accurately predicted. Furthermore it is noted that the Overall Survival (OS) and the Event-Free Survival (EFS) are significantly different between the single (together with the wildtype) and the double mutant samples, as illustrated in Figure 2A-B. It is clear that the samples with double mutations not only have a more favorable risk than the single mutants or samples with a wildtype for CEBPA, but also have a distinct GEP.

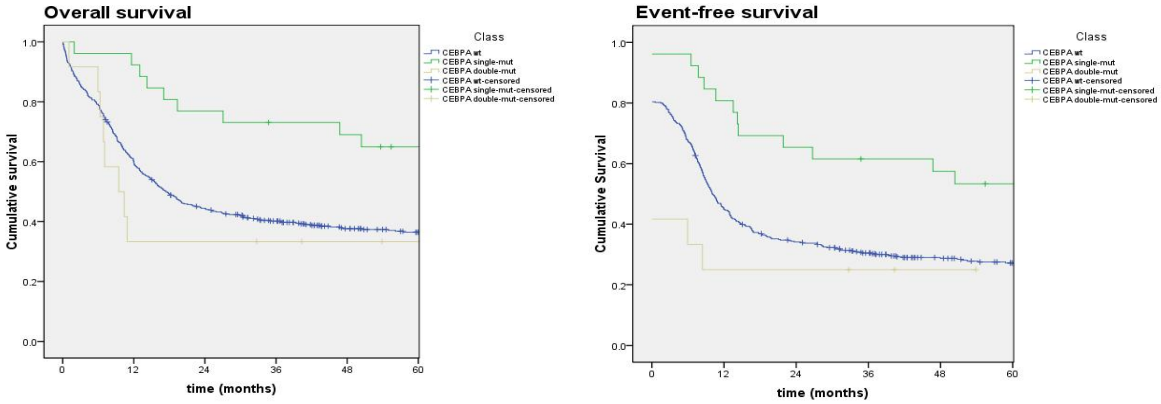


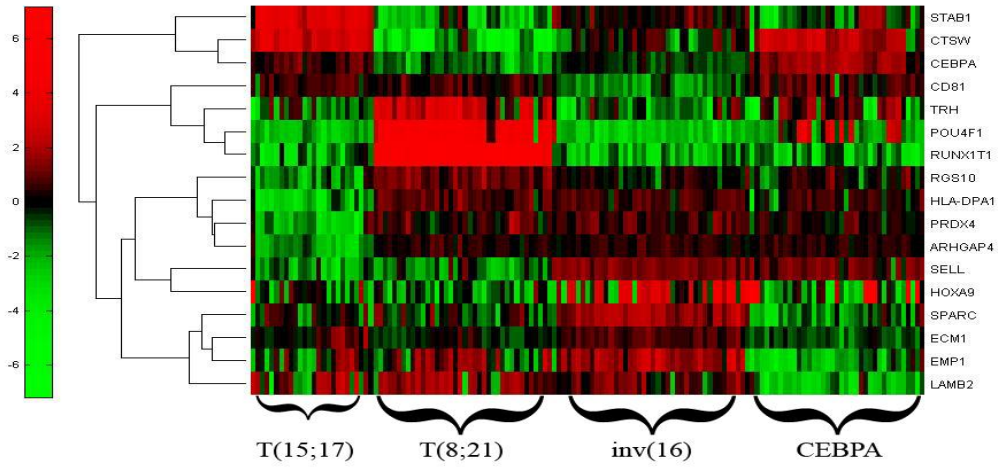
Figure 2. Overall survival and event-free survival.

A. Overall survival among CEBPA$^{double\text{-}mut}$ vs. CEBPA$^{single\text{-}mut}$ vs. CEBPA^{wt}, Log rank test, pooled: p=0.011
B. Event-free survival among CEBPA$^{double\text{-}mut}$ vs. CEBPA$^{single\text{-}mut}$ vs. CEBPA^{wt}, Log rank test, pooled: p=0.008

3.1.3 Interpretation

In addition to determining the prediction accuracy of the algorithm, the interpretation of the retained predictors is an important part of the analysis. Previous research by Kohlmann A. (2003) has also tried to discriminate the defined subtypes (CEBPA excluded). Using microarray data, they selected 23 genes which could accurately classify the subtypes. We have extracted the gene expression profiles of these particular genes from our own AML samples. Similar to the work of Kohlmann, we performed clustering on these genes and plotted them in a heatmap where the colors indicate if the gene was up or down-regulated for that particular individual, as illustrated in Figure 3 (Top). The bottom of Figure 3 contains a subset of the genes with their respective regression coefficients taken from the estimated prediction rule. It is clear that some genes from our prediction rule overlap with the genes of Kohlmann. With the applied penalization we can also see that the regression coefficients of each gene strongly reflect the up or down-regulated tendency of that specific class.

We should note that the retained predictors are not always truly explanatory for the underlying mechanism, such as leukemogenesis. For instance, one well known chromosomal aberration that is recurrent in AML, is the INV(16): the inversion of a part of chromosome 16 results in a fusion protein named CBFB-MYH11. Due to the fusion, the expression levels of MYH11 is substantially increased when compared to other subtypes, thus is a very important biological marker in the diagnosis for this particular subtype. Many classification algorithms with variable selection based on differential expression would automatically select this gene. This is not always the case when our algorithm is applied. The Lasso as well as the Group Lasso automatically selects one predictor if there exists a group of pair-wise correlated and sets all remaining predictors to zero. This could very well be the case for MYH11.



	Other	T(15;17)	T(8;21)	INV(16)	CEBPa	Gene
200665_s_at	-6.62E-06	-4.68E-07	-3.00E-06	1.08E-05	-6.94E-07	SPARC
200675_at	0.000442	8.25E-07	-3.19E-06	-0.00019	-0.00025	CD81
204039_at	-0.00096	0.000151	-0.00062	-0.00068	0.002107	CEBPA
204150_at	-7.40E-05	0.000173	-6.08E-05	-5.21E-05	1.41E-05	STAB1
204563_at	-0.00015	-0.00011	-0.00013	0.000199	0.000194	SELL
205529_s_at	-0.0001	-4.91E-05	0.000257	-9.36E-05	-7.36E-05	RUNX1T1
206940_s_at	-4.49E-05	-2.83E-05	0.000163	-7.21E-05	-1.71E-05	POU4F1
211990_at	-0.00012	-0.00022	0.000108	-1.26E-05	0.000243	HLA-DPA1

Figure 3: (Top) Clustergram: Clustered genes, colors of the cells relate to up- or down regulation of the gene for that particular sample: Green indicates down regulation, Red indicated up regulation. **(Bottom) Regression coefficients:** Regression coefficients for each gene per class.

Previous work by Wunderlich *et al.* (2006), has shown that other genes, such as SPARC and EMP1, are highly correlated with MYH11 in INV(16) patients. Figure 4 illustrates that the probe sets for these genes are significantly up regulated for the INV(16) patients. These probe sets also belong to the top 20 of highest up-regulated genes compared to other groups (Supplementary S5). From the given data, we can conclude that the imposed group structure on the beta matrix $\tilde{\beta}$ results in less retained predictors and also an improved prediction accuracy compared to the Lasso. In addition, many genes related to the retained predictors (Supplementary S1) have been previously associated with leukemogenesis. One example, are the HOXA9 and TRIB1 genes, which are known to be dysregulated in AML samples, and have been identified as cooperative genes together with MEIS1 (Röthlisberger *et al.*,2007; Jin *et al.*, 2007). As stated, the retained genes should not be seen as explanatory, however can be used as a start-off point for further research.

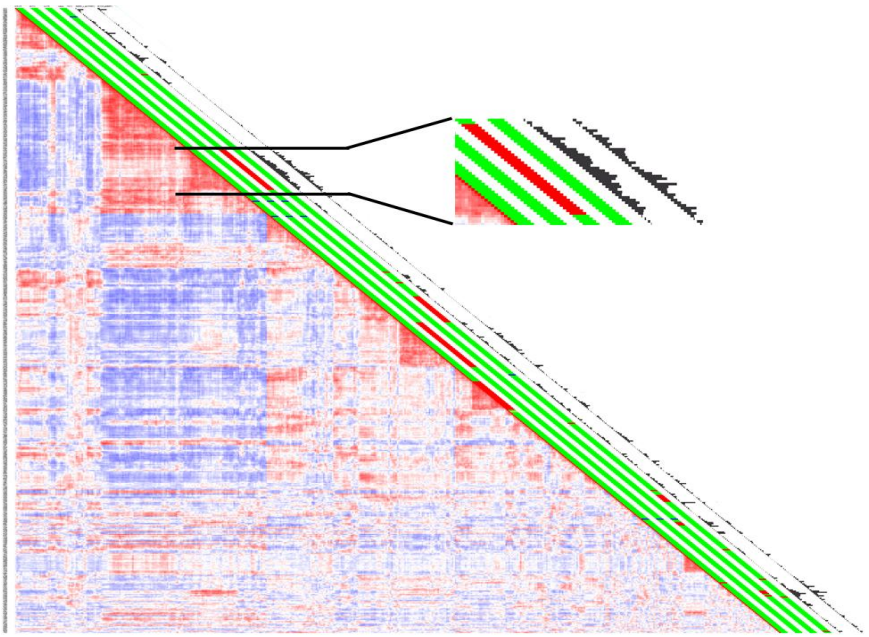


Figure 4: SPARC and EMP1 are elevated in the INV(16) subgroup. Correlation view of the 531 AML patients. Colors of the cells relate to the pair wise Pearson's correlation coefficient values: Red indicates higher positive and blue indicates higher negative correlation between samples. INV(16) aberration status is indicated by the third row next to each tumor (red, mutant; green, wild-type). Histograms next to each tumor indicates the expression levels SPARC and EMP1 respectively, and shows a significant elevated expression for the INV(16) samples.

3.2 Classification case 2: NPM1 and FLT3ITD mutations

	AML cohort 1 (n=261)	AML cohort 2 (n=268)
NPM1-/FLT3ITD-	149(57%)	160(60%)
NPM1+/FLT3ITD-	44(17%)	32(12%)
NPM1-/FLT3ITD+	28(11%)	33(12%)
NPM1+/FLT3ITD+	40(15%)	43(16%)

Table 4: Distribution of the AML samples over the predefined classes.

3.2.1 Background and Classification objective

Mutations in the gene nucleophosmin1 (NPM1) are among one of the most recurrent molecular abnormalities in AML. NPM1 is predominantly found in the nucleolus and is thought to be an important molecular chaperone protein for ribosomal proteins through the cell membrane. Disruption of NPM1, results in the dislocation of NPM1 to the cytoplasm. It has been observed that NPM1 mutations frequently coincide with fms-like tyrosine kinase-3 internal tandem duplication (FLT3ITD) mutations. An additional observation is that NPM1 mutations frequently occur in patients with a normal karyotype and that the dislocation of NPM1 to the cytoplasm leads to its inexertion of its primary function. It has been debated that NPM1 mutation may be an early event in

leukemogenesis. The NPM1 mutation has been associated with as a favorable prognostic value in regard to OS and EFS.

FLT3 is a receptor tyrosine kinase protein that is situated on the cell membrane, where it activates by the binding of the cytokine FLT3 ligand (FLT3L). Binding of the ligand initiates a cascade of signals through second messengers and is known to play an important role in cell differentiation, survival and proliferation. Frequently FLT3 contains an internal tandem duplication and could contribute to the development of AML. Furthermore, the mutation of FLT3 has been associated with as a poor prognostic value in regard to OS and EFS.

In this classification test case, we classify patients which have the NPM1 mutation alone (NPM1+/FLT3ITD-), FLT3ITD alone (NPM1-/FLT3ITD+), both mutations (NPM1+/FLT3ITD+) and the wild-type for these mutations (NPM1-/FLT3ITD-). Table 4 depicts the distribution of these classes.

		Test set error		Sensitivity	Specificity	Predictive Value	
		Neg	Pos	%	%	Neg	Pos
Case 1							
	Other	6/81	0/180	100	93	100	97
	t(15;17)	0/243	0/18	100	100	100	100
	t(8;21)	0/239	0/22	100	100	100	100
	inv(16)	0/238	0/23	100	100	100	100
	CEBPa	0/243	6/18	67	100	98	100
Case 2							
	Other	23/119	7/160	96	81	93	87
	NPM1+/FLT3ITD-	17/237	9/32	72	93	96	58
	NPM1-/FLT3ITD+	6/236	23/33	30	97	91	63
	NPM1+/FLT3ITD+	10/226	17/43	60	96	93	72

Table 5 Prediction outcomes: The following calculations were used for evaluation measures: sensitivity=true positives/(true positive + false negatives), specificity=true negatives/(true negatives + false positives), positive predictive value=true positives/(true positives + false positives), negative predictive value=true negatives /(true negatives + false negatives)

3.2.2 Results

The global test determined that the given genes are significantly related to the outcomes ($p < 0.0001$). With 5-fold cross-validation, we have determined the optimal regularization parameter ($\lambda = 70$) as illustrated in Figure 5. The model retained 110 probe sets (Supplementary S2). For the Lasso penalization we determined the optimal regularization parameter to be 10 with 152 retained probe sets. Using the Group Lasso we can substantially decrease the number of retained predictors for similar prediction accuracy when compared to the Lasso. The Group Lasso falsely classifies 57 samples whereas the Lasso falsely classifies 62 samples (Table 5 vs. Supplementary S6). We took cation not only to compare

the misclassifications, as it is shown that based on the average quadratic loss function the two test errors were quite similar (0.18314 vs. 0.18995).

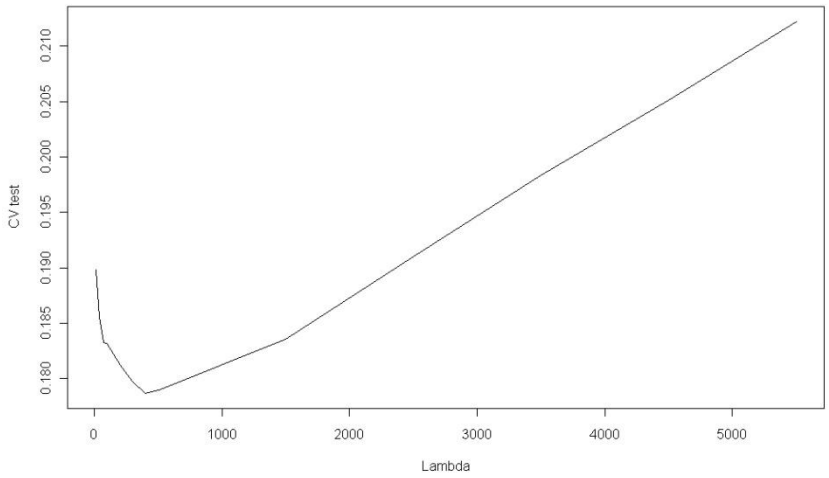


Figure 5 Estimated test error curve based on 5-fold cross-validation

Previous classification work has shown that the mutations in NPM1 are strongly associated with a discriminative HOX-signature (Verhaak *et al.,* 2005; Alcalay *et al.*, 2005). Indeed, our prediction rule has indicated that the presence of the HOXA9 and HOXB3 genes has a strong impact on the classification of NPM1+. A relatively high number of AML cases were falsely classified as having the NPM1 mutation. This could have several reasons: (i) many false positives contained an 11q23 abnormality, which is in line with the affected mixed lineage leukemia (MLL) protein as an important HOX gene expression regulator (Verhaak *et al.*, 2005) (ii) it has been further noted that some subgroup having the FLT3ITD mutation, also exhibit a strong HOX gene expression dysregulation.

The major classification problems stem from the tumors containing the FLT3ITD abnormality. Samples harboring the abnormality can only be moderately classified as indicated in Table 5, possibly due to the following reasons: (i) the cells containing a FLT3ITD abnormality do no diffuse through the whole bone marrow culture. The selection of an appropriate amount of oncogenic cells for correct classification is based on probability, and samples containing a low number of these cells do not have a strong discriminative expression signature (ii) a subgroup of samples harboring the FLT3ITD abnormality behaves differently from the rest.

We can conclude that determining samples with a FLT3ITD abnormality is difficult based on their GEP alone. Most of the NPM1-/FLT3ITD+ samples are falsely classified as wild-type (NPM1-/FLT3ITD-). This is in line with the observation that some if not most of the NPM1-/FLT3ITD+ samples exhibit a weak distinctive GEP. The same holds for the NPM1+/FLT3ITD+ samples which are mostly misclassified as NPM1+/FLT3ITD-, and vice versa. It seems that the

lack of a discriminative FLT3ITD expression signature makes it difficult to concurrently predict all classes with a high accuracy.

3.2.3 Interpretation

The retained predictors from our model application to AML, show an affinity for ribosomal, heatshock, immunoglobulin and HOX proteins. Many genes in the gene expression signature are related to processes of cellular stress, inflammation response and DNA repair mechanisms. The large number of ribosomal genes present in the signature could be due: (i) DNA repair or cell homeostasis mechanisms are activated in the response to the abnormalities arising from oncogenesis (ii) a mutation in the NPM1 gene can result in the dislocation of the protein from the nucleolus to the cytoplasm. The protein is known as a chaperon protein for the ribosomes; however the results could indicate that it may also be involved in the construction of ribosomes, although this is highly speculative.

4. Discussion

The aim of this study was to develop a sparse multi-class classification model based on the Group Lasso in multinomial logistic regression. To create such an algorithm, we have developed a new group structure based on the beta matrix. This group structure facilitates the selection of an entire predictor. We have demonstrated that the prediction accuracy is similar to that of the regular Lasso procedure, yet with less predictors. To illustrate that our approach is effective we have applied the algorithm on microarray gene expression data of a large cohort of well characterized AML patients. Not only have shown that the Group Lasso achieves good prediction accuracy, but also that it obtains a sparse prediction rule containing many previously identified putative cancer genes.

We have demonstrated that our algorithm behaves as expected and we would like to make a note that many different group structures can be developed. We expect in the near feature that singular entities in contemporary classification procedures will be readily replaced by group structures, which increase the interpretability of the prediction rule and generate the opportunity to analyze different aspects of the model. As a final remark we would like to conclude that the development of novel group structures could increase the interpretability of the prediction rule, the prediction accuracy, and possibly further our understanding of cancer and its pathogenesis.

Acknowledgements

We are indebted to Peter J.M. Valk, Justine K. Peeters, Erdogan Taskesen (Erasmus Medical Centre, Rotterdam, The Netherlands) who provided us with data and biologic technical support. We are grateful to Marcel J.T. Reinders (Technical University of Delft, Delft, The Netherlands) who provided us with helpful feedback.

References

Alcalay M., Tiacci E., Bergomas R., Bigerna B, Venturini E., Ninardi SP., et al. (2005). Acute myeloid leukemia bearing cytoplasmic nucleophosmin (NPMc+ AML) shows a distinct gene expression profile characterized by up-regulation of genes involved in stem-cell maintenance, *Blood*, Volume 106(3), pp. 899-902

le Cessie S., van Houwelingen JC. (1992), Ridge Estimators in Logistic Regression, *Applied statistics*, Volume 41(1), 191-201

Jin G., Yamazaki Y., Takuwa M., Takahara T., Kaneko K., Kuwata T., Miyata S., and Nakamura T. (2007), Trib1 and Evi1 cooperate with Hoxa and Meis1 in myeloid leukemogenesis, *Blood* **109** (9) (2007), pp. 3998–4005

Goeman JJ., le Cessie S. (2006), A goodness-of-fit test for multinomial logistic regression, *Biometrics,* Volume 62(4), 980-985

Kohlmann A., Schoch C., Schnitther S., Dugas M., Hiddemann W., Kern W., Haferlach T. (2003), Molecular Characterization of Acute Leukemias by Use of Microarray Technology, *Genes, Chromosomes & Cancer*, Volume 37, 396-405

Krishnapuram B., Carin L., Figueiredo MAT., Hartemink AJ. (2005), Sparse Multinomial Logistic Regression: Fast Algorithms and Generalization Bounds, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Volume 27(6), pp. 957-968

Liu DC., Nocedal J. (1989), On the limited memory BFGS method for large scale optimization, *Mathematical programming 45*, 503-528

Meier L., van de Geer S. and Bühlman P (2008), The Group Lasso for logistic regression. *Journal of the Royal Statistical Society B 70(1)*, 53- 71

Park MY., Trevor H. (2007), L_1 Regularization Path Algorithm for Generalized Linear Models, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Volume 69(4) , pp. 659-677(19)

Röthlisberger B., Heizmann M., Bargetzi M., Huber A., TRIB1 overexpression in acute myeloid leukemia, *Cancer Genetics and Cytogenetics*, Volume 176, Issue 1, Pages 58-60

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

Valk PJM., Verhaak RGW., Beijen MA., Erpelinck CAJ., Barjesteh van Waalwijk van Doorn-Khosrovani S., Boer JM., Beverloo HB., Moorhouse MJ., van der Spek PJ., Löwenberg B., Delwel R. (2004), Prognostically Usefule Gene-Expression Profiles in Acute Myeloid Leukemia*, New England Journal of Medicine 350*, 1617-1628

Verhaak RGW., Goudswaard CS., van Putten W., Bijl MA., Sanders MA., Hugens W., Uitterlinden AG., Erpelinck CAJ, Delwel R., Löwenberg B., and Valk PJM (2005)**,** Mutations in nucleophosmin *(NPM1)* in acute myeloid leukemia (AML): association with other gene abnormalities and previously established gene expression signatures and their favorable prognostic significance, *Blood 106(12)*, 3747-3754

Wunderlich M., Krejci O., Wei J., Mulloy JC. (2006), Human $CD34^{+}$ cells expressing the inv(16) fusion protein exhibit myelomonocytic phenotype with greatly enhanced proliferative ability, *Blood*, Volume 108(5), 1690-1697

Wouters BJ., Löwenberg B., Erpelinck-Verschueren CAJ., van Putten WLJ., Valk PJM., and Delwel R. (2009), Double CEBPA mutations, but not single CEBPA mutations, define a subgroup of acute myeloid leukemia with a distinctive gene expression profile that is uniquely associated with a favorable outcome, *Blood* 113(13), 3088-3091

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

Zhu J., Hastie T. (2004), Classification of gene microarrays by penalized logistic regression, *Biostatistics,* Volume 5(3), 427-443

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

Supplementary

S1 Group Lasso: Gene expression signatures of chromosomal aberrations and CEBPa

	Other	T(15;17)	T(8;21)	INV(16)	CEBPa	Gene
Intercept	3.92E-07	-1.16E-07	2.69E-09	-1.36E-07	-1.40E-07	
1553588_at	0.000265	-4.47E-05	9.67E-05	-0.00019	-0.00013	ND3 /// SH3KBP1
1555745_a_at	1.85E-05	-7.99E-06	-9.06E-06	6.77E-06	-8.17E-06	LYZ
200091_s_at	1.34E-06	-6.29E-08	-2.12E-08	-2.38E-08	-1.24E-06	RPS25
200654_at	9.46E-05	0.000146749	-2.97E-05	-6.76E-05	-0.00014	P4HB
200665_s_at	-6.62E-06	-4.68E-07	-3.00E-06	1.08E-05	-6.94E-07	SPARC
200675_at	0.000442	8.25E-07	-3.19E-06	-0.00019	-0.00025	CD81
200748_s_at	0.00014	-1.53E-05	-3.36E-05	3.64E-05	-0.00013	FTH1
200869_at	1.37E-05	-6.24E-06	-1.87E-06	4.40E-06	-1.00E-05	RPL18A
200909_s_at	0.000667	-0.00021971	-8.78E-05	1.06E-05	-0.00037	RPLP2
200920_s_at	0.000374	-4.90E-06	0.000136	-5.89E-05	-0.00045	BTG1
200921_s_at	5.24E-05	-3.21E-06	1.59E-05	-1.44E-05	-5.07E-05	BTG1
201160_s_at	0.000266	7.82E-05	8.42E-05	0.000101	-0.00053	CSDA
201360_at	-1.65E-06	-8.17E-07	-1.39E-05	4.00E-05	-2.37E-05	CST3
201432_at	0.000551	-0.00031515	-0.00012	-0.00083	0.000712	CAT
201669_s_at	2.88E-05	-6.28E-06	1.87E-06	-1.89E-05	-5.50E-06	MARCKS
201720_s_at	3.44E-05	-2.86E-06	-3.02E-05	5.16E-06	-6.51E-06	LAPTM5
201721_s_at	0.000222	1.32E-05	-0.00022	3.67E-05	-5.10E-05	LAPTM5
202241_at	0.000119	-4.72E-05	-1.12E-05	1.94E-05	-8.00E-05	TRIB1
202649_x_at	0.000113	-4.06E-05	-2.65E-06	2.12E-05	-9.13E-05	RPS19
202746_at	-0.00113	6.90E-05	-0.00026	0.000667	0.00065	ITM2A
202902_s_at	-4.04E-05	-2.46E-05	-1.89E-05	6.40E-05	1.98E-05	CTSS
202917_s_at	-4.99E-05	-0.00015906	-7.98E-05	0.000254	3.52E-05	S100A8
203535_at	-4.42E-05	-7.21E-05	-0.0001	0.00036	-0.00014	S100A9
203752_s_at	0.000113	-2.17E-05	1.33E-05	-4.71E-05	-5.75E-05	JUND
203948_s_at	-0.00064	0.000345876	0.000277	0.000239	-0.00022	MPO
203973_s_at	-0.00033	1.15E-05	-2.73E-06	0.000201	0.000116	CEBPD
204039_at	-0.00096	0.000150948	-0.00062	-0.00068	0.002107	CEBPA
204150_at	-7.40E-05	0.000172869	-6.08E-05	-5.21E-05	1.41E-05	STAB1
204563_at	-0.00015	-0.00011174	-0.00013	0.000199	0.000194	SELL
205237_at	0.00015	2.08E-05	-5.01E-05	-0.00034	0.000219	FCN1
205382_s_at	1.07E-05	2.77E-05	-2.69E-06	9.79E-07	-3.66E-05	CFD
205529_s_at	-0.0001	-4.91E-05	0.000257	-9.36E-05	-7.36E-05	RUNX1T1
205683_x_at	-0.00024	4.74E-05	0.000126	1.88E-05	4.93E-05	TPSAB1 /// TPSB2

206111_at	1.88E-05	0.000319665	-0.00048	0.000393	-0.00025	HLA-DRB1
206834_at	-0.00022	4.74E-05	-0.00012	-0.00012	0.000408	HBD
206871_at	-0.00063	0.00030899	0.000333	-1.32E-05	1.40E-06	ELA2
206940_s_at	-4.49E-05	-2.83E-05	0.000163	-7.21E-05	-1.71E-05	POU4F1
207134_x_at	-6.03E-13	2.44E-14	2.19E-13	8.01E-15	8.86E-14	TPSB2
207168_s_at	-0.00022	2.27E-05	-1.25E-05	2.03E-06	0.00021	H2AFY
207741_x_at	-9.34E-05	1.47E-05	4.65E-05	-4.03E-06	3.62E-05	TPSAB1
208306_x_at	-6.19E-05	-0.00013182	4.59E-05	8.33E-05	6.47E-05	HLA-DRB1
209189_at	-2.54E-05	-2.91E-05	-2.07E-05	3.94E-05	3.58E-05	FOS
209312_x_at	-0.00021	-0.00027601	0.000108	0.000165	0.000217	HLA-DRB1
209619_at	-5.49E-07	-9.22E-05	3.23E-05	4.61E-06	5.59E-05	CD74
210084_x_at	-3.29E-05	3.61E-06	1.63E-05	3.75E-06	9.24E-06	TPSAB1
211341_at	-0.00022	-0.00014343	0.000788	-0.00034	-8.64E-05	POU4F1
211709_s_at	-0.0007	0.000365002	0.000105	6.15E-05	0.000167	CLEC11A
211745_x_at	6.46E-05	-1.04E-05	-6.64E-05	3.16E-05	-1.93E-05	HBA1 /// HBA2
211956_s_at	5.92E-05	-1.07E-05	2.28E-06	-8.14E-06	-4.26E-05	EIF1
211990_at	-0.00012	-0.00022121	0.000108	-1.26E-05	0.000243	HLA-DPA1
212099_at	-2.69E-05	5.13E-06	-5.56E-06	5.70E-05	-2.97E-05	RHOB
212560_at	-8.67E-08	1.03E-07	2.24E-07	-4.88E-07	2.49E-07	SORL1
212587_s_at	0.000132	-1.19E-05	1.20E-06	1.76E-05	-0.00014	PTPRC
213515_x_at	0.000245	3.18E-05	-0.00011	-0.0001	-6.40E-05	HBG1 /// HBG2
213737_x_at	0.000457	-8.94E-05	2.14E-05	-0.00054	0.000153	GOLGA9P
214039_s_at	1.32E-05	-2.07E-06	6.58E-07	-3.07E-06	-8.69E-06	LAPTM4B
214651_s_at	0.00019	-7.17E-05	-3.48E-05	-7.62E-05	-7.69E-06	HOXA9
215382_x_at	-0.00017	1.50E-05	7.78E-05	2.33E-05	5.53E-05	TPSAB1
215806_x_at	1.49E-05	1.55E-05	-1.10E-05	-2.20E-05	2.52E-06	TARP /// TRGC2
216474_x_at	-0.00032	5.44E-05	0.000161	1.37E-05	8.77E-05	TPSAB1 /// TPSB2
216920_s_at	8.90E-05	9.27E-05	-6.54E-05	-0.00013	1.67E-05	TARP /// TRGC2
217022_s_at	0.000126	-4.24E-05	4.66E-05	-5.33E-05	-7.66E-05	IGH@
219014_at	-3.30E-06	2.26E-06	-5.82E-06	1.80E-06	5.09E-06	PLAC8
219371_s_at	1.64E-06	-2.73E-05	9.24E-06	0.00014	-0.00012	KLF2
221760_at	-0.00011	-8.86E-05	0.000178	-8.08E-05	0.000103	MAN1A1
221841_s_at	-2.63E-05	-5.32E-05	-5.97E-05	0.000211	-7.21E-05	KLF4
223059_s_at	0.000133	-3.70E-05	1.75E-05	-3.58E-05	-7.79E-05	FAM107B
225262_at	2.80E-05	1.24E-06	-2.24E-06	-2.05E-06	-2.50E-05	FOSL2
225673_at	6.63E-05	-4.24E-05	5.08E-05	1.93E-07	-7.53E-05	MYADM
226131_s_at	2.30E-05	-5.65E-06	4.53E-07	-1.38E-07	-1.78E-05	RPS16
226818_at	-3.95E-16	-5.15E-16	-7.05E-16	6.07E-16	2.64E-16	MPEG1
226876_at	1.69E-16	1.31E-16	-5.96E-16	2.00E-16	-1.58E-16	FAM101B
226905_at	0.000178	0.000145107	-0.00017	9.75E-06	-0.00016	FAM101B
227404_s_at	-0.00018	-0.00012422	-1.10E-05	8.66E-05	0.000224	EGR1

38487_at	-6.83E-05	0.00020396	-7.35E-05	-5.88E-05	-3.45E-06	STAB1

S2 Group Lasso: Gene expression signatures of NPM1 and FLT3ITD mutations

	NPM1-/FLT3ITD-	NPM1+/FLT3ITD-	NPM1-/FLT3ITD+	NPM1+/FLT3ITD+	Gene
Intercept	1.78E-07	8.76E-09	-2.01E-07	1.52E-08	
211983_x_at	-4.19E-06	-1.54E-06	4.64E-06	1.09E-06	ACTG1
224585_x_at	-4.50E-05	-1.65E-05	5.03E-05	1.12E-05	ACTG1
201012_at	-0.000360372	0.000195166	0.000139287	2.65E-05	ANXA1
214575_s_at	-4.97E-05	7.87E-05	8.94E-05	-0.000118497	AZU1
200920_s_at	-3.44E-05	0.000232044	-0.000119983	-7.77E-05	BTG1
201310_s_at	3.70E-05	2.10E-05	-2.96E-05	-2.84E-05	C5orf13
201432_at	-0.000198258	0.000191504	-0.000102705	0.000109614	CAT
211922_s_at	-1.68E-05	2.88E-05	-6.94E-06	-5.05E-06	CAT
209543_s_at	0.000146937	-4.73E-05	-7.24E-05	-2.73E-05	CD34
209555_s_at	-0.000339683	9.66E-05	-3.05E-05	0.000273657	CD36
209835_x_at	0.000100106	-2.50E-05	-2.70E-05	-4.81E-05	CD44
209619_at	-3.81E-05	-4.50E-05	0.000147403	-6.44E-05	CD74
201029_s_at	9.68E-07	-8.10E-06	-3.56E-07	7.49E-06	CD99
208727_s_at	1.79E-05	1.78E-06	-2.57E-05	5.99E-06	CDC42
211709_s_at	5.26E-09	1.93E-09	-3.68E-09	-3.30E-09	CLEC11A
205624_at	-0.000144824	-1.52E-05	-0.000110138	0.000270098	CPA3
201160_s_at	8.77E-05	6.74E-05	-3.35E-05	-0.00012162	CSDA
1553297_a_at	0.000319123	-0.000210977	-2.77E-05	-8.04E-05	CSF3R
201360_at	-4.69E-05	0.000104419	-4.26E-05	-1.49E-05	CST3
205898_at	-2.06E-05	6.33E-05	-3.43E-05	-8.42E-06	CX3CR1
201041_s_at	4.55E-05	4.79E-05	-6.38E-05	-2.96E-05	DUSP1
211937_at	4.20E-05	8.60E-06	2.67E-05	-7.73E-05	EIF4B
206871_at	8.99E-05	-7.19E-05	1.07E-05	-2.87E-05	ELA2
221804_s_at	-4.72E-12	-2.03E-12	3.87E-12	1.10E-12	FAM45A
200019_s_at	1.37E-07	-1.33E-07	-4.87E-08	4.04E-08	FAU
218454_at	0.000148374	-0.000158047	-3.91E-05	4.88E-05	FLJ22662
206674_at	-3.02E-08	2.42E-08	2.66E-08	-2.02E-08	FLT3
200748_s_at	-6.84E-06	7.38E-05	-5.64E-06	-6.13E-05	FTH1
212788_x_at	1.25E-05	-4.55E-05	1.15E-06	3.19E-05	FTL
212581_x_at	-3.75E-07	-3.54E-07	4.32E-07	2.57E-07	GAPDH
200648_s_at	-4.94E-12	6.28E-12	7.94E-12	-8.38E-12	GLUL
215001_s_at	-3.87E-05	4.21E-05	9.06E-05	-9.40E-05	GLUL
205349_at	-4.52E-05	-3.24E-05	5.38E-05	2.37E-05	GNA15
208798_x_at	-8.07E-06	-5.96E-06	4.95E-06	9.08E-06	GOLGA8A
210425_x_at	-5.20E-05	-4.31E-05	2.74E-05	6.78E-05	GOLGA8A
208886_at	0.000159109	-0.000230503	0.000268639	-0.000197244	H1F0

207168_s_at	4.91E-05	-5.77E-05	0.000186878	-0.000178307	H2AFY
209458_x_at	-1.51E-06	2.66E-06	4.43E-05	-4.55E-05	HBA1
217232_x_at	-3.67E-05	8.36E-05	1.57E-05	-6.26E-05	HBB
214290_s_at	-5.65E-06	1.46E-05	-0.000397119	0.000387664	HIST2H2AA3
215313_x_at	1.52E-05	-1.13E-05	-1.33E-05	9.45E-06	HLA-A
201137_s_at	-7.03E-06	-4.67E-05	4.92E-05	4.48E-06	HLA-DPB1
206111_at	-9.89E-06	3.41E-05	-9.64E-06	-1.46E-05	HLA-DRB1
208306_x_at	-1.36E-05	-7.12E-05	0.000111589	-2.69E-05	HLA-DRB1
209312_x_at	9.11E-06	-0.000162586	0.000170839	-1.74E-05	HLA-DRB1
215193_x_at	-1.71E-06	-9.58E-06	1.05E-05	7.80E-07	HLA-DRB1
214651_s_at	-0.000244243	6.22E-05	-0.000241893	0.00042393	HOXA9
228904_at	-0.000514691	0.00017295	9.99E-05	0.000241824	HOXB3
1557910_at	-6.82E-05	2.92E-05	7.09E-05	-3.20E-05	HSP90AB1
200799_at	-6.46E-05	9.85E-05	0.00013247	-0.000166403	HSPA1A
201315_x_at	5.02E-14	-6.20E-13	4.21E-13	-2.00E-13	IFITM2
201163_s_at	0.000158966	6.45E-05	-7.47E-05	-0.000148747	IGFBP7
217022_s_at	2.88E-05	-1.41E-06	-0.000115641	8.82E-05	IGH@
221651_x_at	6.12E-06	-3.11E-06	-2.85E-05	2.55E-05	IGK@
224795_x_at	3.13E-05	-6.52E-06	-0.000123394	9.86E-05	IGK@
215121_x_at	2.62E-05	1.71E-05	-2.32E-05	-2.00E-05	IGL@
202746_at	7.73E-05	-4.83E-05	-3.75E-05	8.52E-06	ITM2A
201464_x_at	1.16E-05	-8.42E-06	-1.27E-05	9.55E-06	JUN
203752_s_at	1.72E-05	-2.92E-05	-1.89E-05	3.09E-05	JUND
219371_s_at	0.000287501	0.000246003	-0.000303261	-0.000230227	KLF2
214039_s_at	-1.47E-05	-1.01E-05	9.97E-06	1.48E-05	LAPTM4B
201105_at	-1.29E-06	1.56E-06	2.41E-06	-2.69E-06	LGALS1
200923_at	-0.000374201	-0.000101345	0.000113287	0.00036226	LGALS3BP
234512_x_at	1.05E-05	-1.84E-05	3.63E-05	-2.84E-05	LOC728179
1555745_a_at	4.65E-05	-1.10E-05	-0.000101175	6.56E-05	LYZ
222670_s_at	-6.77E-06	1.34E-05	-3.20E-05	2.54E-05	MAFB
1558678_s_at	0.000107329	-2.05E-05	-0.000125333	3.85E-05	MALAT1
203949_at	5.86E-06	-9.81E-05	5.01E-05	4.22E-05	MPO
204438_at	-4.68E-05	-0.000129185	0.000140988	3.50E-05	MRC1
212185_x_at	2.90E-05	-1.44E-05	-1.29E-05	-1.68E-06	MT2A
225344_at	-2.52E-05	-5.45E-05	7.62E-05	3.48E-06	NCOA7
234989_at	3.89E-05	3.22E-05	-7.74E-05	6.42E-06	NCRNA00084
1553588_at	-1.30E-05	-7.42E-05	1.34E-05	7.39E-05	ND3
223217_s_at	5.63E-05	0.000129459	-0.000100436	-8.53E-05	NFKBIZ
223218_s_at	1.42E-05	6.24E-05	-2.85E-05	-4.82E-05	NFKBIZ
212240_s_at	-8.39E-05	-3.79E-05	6.22E-05	5.97E-05	PIK3R1
219014_at	4.43E-05	-8.22E-05	3.88E-05	-8.34E-07	PLAC8

214146_s_at	-0.000105048	0.000121205	-9.54E-05	7.93E-05	PPBP
202130_at	1.28E-05	-8.12E-06	-5.31E-06	6.36E-07	RIOK3
224930_x_at	2.25E-05	-8.25E-05	0.000300059	-0.000239988	RPL7A
200032_s_at	6.51E-05	-1.52E-05	-5.50E-05	5.05E-06	RPL9
200909_s_at	0.000114	-5.47E-05	-4.18E-05	-1.76E-05	RPLP2
200817_x_at	4.12E-05	-1.81E-05	-3.28E-05	9.74E-06	RPS10
217753_s_at	4.61E-05	-4.84E-05	-1.69E-05	1.91E-05	RPS26
200099_s_at	8.55E-05	3.19E-05	-4.71E-05	-7.03E-05	RPS3A
201909_at	5.16E-05	4.02E-05	3.73E-06	-9.55E-05	RPS4Y1
203408_s_at	5.18E-05	6.07E-06	-5.99E-05	2.03E-06	SATB1
204563_at	-4.45E-05	7.41E-05	3.10E-05	-6.07E-05	SELL
201427_s_at	-4.49E-05	-0.000125351	-1.74E-05	0.000187606	SEPP1
221269_s_at	0.000358997	-1.37E-05	-0.000360254	1.49E-05	SH3BGRL3
212826_s_at	0.000383135	-0.000305564	-2.73E-05	-5.03E-05	SLC25A6
201663_s_at	-0.000283041	0.000149964	-2.48E-05	0.000157812	SMC4
201664_at	-0.000386044	0.000244129	-5.00E-05	0.000191897	SMC4
204466_s_at	0.000151217	-4.37E-05	-6.63E-05	-4.13E-05	SNCA
212560_at	7.84E-05	2.54E-05	0.000161574	-0.00026544	SORL1
215806_x_at	-3.43E-05	-5.00E-05	7.15E-05	1.28E-05	TARP
216920_s_at	-0.000110091	-0.000114984	0.000186388	3.87E-05	TARP
201666_at	1.90E-05	-1.69E-05	-9.62E-06	7.46E-06	TIMP1
205683_x_at	6.62E-05	-3.48E-05	2.28E-05	-5.41E-05	TPSAB1
215382_x_at	6.01E-05	-3.36E-05	1.86E-05	-4.52E-05	TPSAB1
216474_x_at	2.40E-05	-1.41E-05	8.76E-06	-1.86E-05	TPSAB1
209118_s_at	4.09E-06	1.20E-06	-1.36E-06	-3.93E-06	TUBA1A
201009_s_at	-4.04E-05	0.000142586	-5.03E-05	-5.19E-05	TXNIP
204620_s_at	-0.00019742	4.76E-05	5.80E-05	9.20E-05	VCAN
215646_s_at	-2.92E-05	1.30E-05	6.38E-06	9.86E-06	VCAN
221731_x_at	-0.000173184	4.45E-05	5.99E-05	6.89E-05	VCAN
201426_s_at	-7.98E-08	-4.86E-09	6.44E-08	1.83E-08	VIM
200670_at	-0.00015436	1.57E-05	0.000129817	8.85E-06	XBP1
213655_at	-1.59E-08	-3.17E-08	3.99E-09	4.34E-08	YWHAE
201368_at	8.99E-05	-0.000161074	-5.93E-05	0.000130444	ZFP36L2

S3 Lasso: Gene expression signatures of chromosomal aberrations and CEBPa

	Other	T(15;17)	T(8;21)	INV(16)	CEBPa	Gene
Intercept	9.75E-06	-1.16E-06	4.67E-07	-3.58E-06	-5.51E-06	
1553588_at	0	0	0	-0.00021	-0.00015	ND3 /// SH3KBP1
1555745_a_at	0.000357	0	0	0	-6.50E-05	LYZ
200091_s_at	2.45E-06	0	0	0	0	RPS25
200654_at	0	0	0	0	-0.00012	P4HB
200665_s_at	0	0	0	0.000547	0	SPARC
200675_at	0.00239	0	0	0	0	CD81
200748_s_at	0.000302	0	0	0	-0.00016	FTH1
200869_at	4.90E-05	0	0	0	0	RPL18A
200909_s_at	0.003134	0	0	0	0	RPLP2
200920_s_at	0	0	0	0	-0.00054	BTG1
201160_s_at	0	0	0	0	-0.00419	CSDA
201360_at	0	0	0	1.58E-05	0	CST3
201432_at	0	0	0	-0.00216	5.85E-05	CAT
201858_s_at	0.000267	0	0	0	0	SRGN
201909_at	0	0	0	0	0.000772	RPS4Y1
202081_at	-0.00015	0	0	0	0	IER2
202649_x_at	0.000968	0	0	0	-5.98E-05	RPS19
202746_at	-0.00507	0	0	0.001124	0	ITM2A
202859_x_at	0	0	0.000867	0	0	IL8
202902_s_at	0	0	0	4.13E-05	1.64E-04	CTSS
202917_s_at	0	0	0	8.15E-05	0	S100A8
203305_at	0.000649	0	0	0	-0.00099	F13A1
203373_at	8.90E-05	0	0	0	0	SOCS2
203535_at	0	0	0	0.002361	0	S100A9
203752_s_at	0.001777	0	0	0	0	JUND
203948_s_at	-0.0023	0	0	0	0	MPO
203949_at	0	0	0	0	-0.00161	MPO
203973_s_at	-0.00163	0	0	0	0	CEBPD
204039_at	0	0	0	0	0.009463	CEBPA
204304_s_at	0	0	0	0	0.001213	PROM1
204563_at	0	0	0	0.001144	0.001079	SELL
204670_x_at	0	0	0	0	0.000318	HLA-DRB1
205237_at	0	0	0	-0.00168	0	FCN1
205382_s_at	0	0	0	0	-1.06E-05	CFD
205529_s_at	0	0	0.00245	0	0	RUNX1T1

206111_at	0	0	-0.00152	0	-0.00055	HLA-DRB1
206834_at	0	0	0	0	0.001443	HBD
206871_at	-0.0023	0	9.06E-06	-0.00012	0	ELA2
207168_s_at	-0.00051	0	0	0	2.69E-09	H2AFY
208306_x_at	0	-0.00057	0	0	0	HLA-DRB1
209069_s_at	0.000212	0	0	0	0	H3F3A
209189_at	0	0	0	3.55E-05	0	FOS
209312_x_at	-0.00142	0	0	0	0	HLA-DRB1
209619_at	0	-0.00106	0	0	0	CD74
210140_at	0	0.001121	0	0	0	CST7
210997_at	0	0.000601	0	0	0	HGF
211341_at	0	0	0.004471	0	0	POU4F1
211709_s_at	-0.00383	0	0	0	0	CLEC11A
211745_x_at	0.000162	0	0	0	0	HBA1 /// HBA2
211956_s_at	0.000537	0	0	0	0	EIF1
211990_at	0	0	0	0	0.001528	HLA-DPA1
212085_at	0	0	0	0.000555	0	SLC25A6
212099_at	0	0	0	8.94E-04	0	RHOB
212560_at	0	1.03E-07	0	0	0	SORL1
212587_s_at	0	0	0	0	-0.00076	PTPRC
213515_x_at	0.000929	0	0	0	0	HBG1 /// HBG2
213737_x_at	0.000994	0	0	-0.00145	0	GOLGA9P
214039_s_at	2.46E-04	0	0	0	0	LAPTM4B
214651_s_at	0.000962	0	0	0	0	HOXA9
216248_s_at	0	0	0	2.02E-05	0	NR4A2
216474_x_at	-0.00319	0	0	0	0	TPSAB1 /// TPSB2
217022_s_at	2.17E-05	0	0	0	0	IGH@
219014_at	0	0	0	0	2.35E-05	PLAC8
219371_s_at	0	0	0	0.000789	-4.17E-05	KLF2
220532_s_at	0.000652	0	0	0	0	TMEM176B
221760_at	-0.00011	0	0	0	0.000103	MAN1A1
221841_s_at	0	0	0	0.002957	0	KLF4
223059_s_at	0.003553	0	0	0	0	FAM107B
225262_at	0.000908	0	0	0	-0.00152	FOSL2
225673_at	0.000816	0	0	0	-0.00015	MYADM
226818_at	0	0	0	1.74E-05	0	MPEG1
226905_at	0	5.69E-05	0	0	0	FAM101B
227404_s_at	-0.00036	0	0	0	0.001173	EGR1
229307_at	0	0	0	0	0.001112	ANKRD28
234989_at	0	0	0	0	-8.36E-05	NCRNA00084
38487_at	0	0.003864	0	0	0	STAB1

S4 Lasso: Gene expression signatures of NPM1 and FLT3ITD mutations

	NPM1-/FLT3ITD-	NPM1+/FLT3ITD-	NPM1-/FLT3ITD+	NPM1+/FLT3ITD+	Gene
Intercept	2.69E-05	-8.92E-07	-4.02E-05	1.42E-05	
211983_x_at	-0.000549406	0	0	0	ACTG1
201012_at	-0.001924298	0	0	0	ANXA1
214575_s_at	0	0	3.45E-06	-0.000498458	AZU1
202391_at	0	0	-0.000181342	0	BASP1
200920_s_at	0	0.001596801	0	0	BTG1
209301_at	0.000861618	0	-0.000203814	0	CA2
200953_s_at	0	-0.000141761	0	0	CCND2
201743_at	0	0	0	4.15E-05	CD14
209543_s_at	0.000553516	0	0	0	CD34
209555_s_at	-0.001964638	0	0	0.00014582	CD36
228766_at	-7.96E-05	0	0	0	CD36
209835_x_at	0.000143441	0	0	0	CD44
209619_at	0	0	0.000385769	0	CD74
208727_s_at	0	0	-0.000743477	0	CDC42
205382_s_at	0	0.000247366	0	0	CFD
211709_s_at	0.000644951	0	0	0	CLEC11A
201560_at	0	-0.004186131	0	0	CLIC4
205624_at	0	0	0	0.000982105	CPA3
201160_s_at	0	0.000477698	0	-0.002703532	CSDA
1553297_a_at	0.00215887	0	0	0	CSF3R
201360_at	0	0.00028234	0	0	CST3
205653_at	0	0	0	0.000309869	CTSG
202902_s_at	0	-0.000875538	0	0.000670705	CTSS
205898_at	0	0.000961345	0	0	CX3CR1
208151_x_at	0	0	0	-0.002207349	DDX17
205033_s_at	0	6.98E-05	0	0	DEFA1
207269_at	0	0.000421954	0	0	DEFA4
1566363_at	0.000552686	0	0	0	DNTT
211937_at	0	0	0	-4.23E-05	EIF4B
205767_at	0	0.000166241	0	-0.000626061	EREG
221804_s_at	0	0	0.001082107	0	FAM45A
221766_s_at	0	-6.08E-05	0	0	FAM46A
201540_at	0	0	0	-0.001138038	FHL1
218454_at	0	-0.001630132	0	0	FLJ22662
200859_x_at	0	0	0	-0.000493235	FLNA
202768_at	0	0	0	-0.000694321	FOSB
200748_s_at	0	0.00031813	0	-0.000855146	FTH1

212788_x_at	0	-0.001064058	0	0	FTL
215001_s_at	0	0	2.78E-05	-0.000392583	GLUL
205349_at	0	0	0.00123184	0	GNA15
210425_x_at	0	0	0	0.000358595	GOLGA8A
208886_at	0	0	0.000945981	0	H1F0
207168_s_at	0	0	0.000328776	0	H2AFY
213828_x_at	0	0	-0.001700515	0	H3F3A
209458_x_at	0	0	0	-0.00046558	HBA1
214414_x_at	4.41E-05	0	0	0	HBA1
209116_x_at	0	0	0.000135244	0	HBB
217232_x_at	-0.000439071	0.000136679	0	0	HBB
204419_x_at	0	0	0	0.000151276	HBG1
240336_at	0	-0.000137841	0	0.001521039	HBM
214290_s_at	0	0	-0.001162443	0.000782118	HIST2H2AA3
211911_x_at	0.000190435	0	0	0	HLA-B
201137_s_at	0	0	0.000782876	0	HLA-DPB1
206111_at	0	0.000587791	0	0	HLA-DRB1
208306_x_at	0	0	0.000736662	0	HLA-DRB1
209312_x_at	0	-0.000737516	0	0	HLA-DRB1
215193_x_at	0	0	0.000476996	0	HLA-DRB1
208808_s_at	0	0	0.001337137	0	HMGB2
200943_at	0	0	0.000404683	0	HMGN1
214651_s_at	0	0	0	0.002911237	HOXA9
228904_at	-0.00326353	0	0	0	HOXB3
200799_at	0	0.000674539	0	-0.001735493	HSPA1A
211936_at	-0.000995856	0	0	0	HSPA5
201163_s_at	0.00104462	0	0	0	IGFBP7
221671_x_at	0	0	0	0.000212086	IGK@
224795_x_at	0	0	-0.002148643	0	IGK@
211945_s_at	0	0	-0.00212093	0	ITGB1
202746_at	0.000974254	0	0	0	ITM2A
201464_x_at	2.73E-05	-5.10E-05	0	0	JUN
203752_s_at	0	0	0	0.00094462	JUND
219371_s_at	0.000110546	0.000328912	-0.000238904	-0.000192295	KLF2
201553_s_at	0.000612592	0	0	0	LAMP1
201105_at	0	0	0	-0.000623569	LGALS1
200923_at	-0.000743235	0	0	0.002400105	LGALS3BP
238893_at	0	0	0	3.89E-05	LOC338758
236488_s_at	0	0	0	0.000244434	LOC642711
226789_at	0.000751003	0	0	0	LOC647121
1555745_a_at	0	0	-0.00028733	0	LYZ

213975_s_at	0	-6.02E-05	0	0	LYZ
222670_s_at	0	0	-2.78E-06	0	MAFB
36711_at	0	-0.000587123	0.000274669	0	MAFF
1558678_s_at	0.000365192	0	-0.000235455	0	MALAT1
201669_s_at	0	0	-0.001111415	0	MARCKS
203948_s_at	0	0	-6.91E-05	0	MPO
203949_at	0	-0.00061646	0	0	MPO
224356_x_at	0.000552355	0	0	0	MS4A6A
212185_x_at	0.001279688	0	0	0	MT2A
211445_x_at	0	-7.71E-06	0	0	NACAP1
208752_x_at	0	0.00149698	0	0	NAP1L1
225344_at	0	0	0.002102823	0	NCOA7
234989_at	4.62E-05	0	-0.000584575	0	NCRNA00084
201502_s_at	0	0	-0.000544236	0.000551416	NFKBIA
223217_s_at	0	0.00016798	0	0	NFKBIZ
223218_s_at	0	0.000788183	0	-0.00054332	NFKBIZ
221501_x_at	0	0	-0.000545477	0	NPIP
226880_at	0	0.000355529	0	0	NUCKS1
208690_s_at	-0.000240437	0	0	0	PDLIM1
206390_x_at	0	0.000391525	0	0	PF4
201118_at	0	0	7.17E-05	0	PGD
212240_s_at	-0.000183918	0	0	0	PIK3R1
211978_x_at	-0.000843402	0	0	0	PPIA
207341_at	0	0	4.49E-05	0	PRTN3
211600_at	0	0.000247919	0	0	PTPRO
212099_at	0	0	-0.001687105	0	RHOB
200088_x_at	-8.62E-05	0	0	0	RPL12
216570_x_at	0.001975063	0	0	0	RPL29P4
200674_s_at	0	0.000333418	0	0	RPL32
224930_x_at	0	0	0.001783615	-0.000987708	RPL7A
200032_s_at	0	0	-0.001311943	0	RPL9
201033_x_at	0	-0.000414652	0	0	RPLP0
211720_x_at	0	-0.001238522	0	0	RPLP0
214167_s_at	0	-9.32E-05	0	0	RPLP0
200909_s_at	0.000864384	0	0	0	RPLP2
214003_x_at	0	-0.000528779	0	0	RPS20
200926_at	0	0	0	0.00038619	RPS23
217753_s_at	0	-0.000810844	0	0	RPS26
200099_s_at	0	0	0	-0.000709369	RPS3A
201909_at	0	0	0	-0.000928506	RPS4Y1
214317_x_at	0	0	-0.000926541	0	RPS9

200872_at	3.04E-05	0	0	0	S100A10
230333_at	0	-0.000118766	0	0	SAT
204563_at	-0.000121405	9.69E-05	8.74E-06	-8.18E-06	SELL
201427_s_at	0	0	0	0.001167497	SEPP1
201586_s_at	-0.001128205	0	0	0	SFPQ
221269_s_at	0.001087794	0	-0.001126426	0	SH3BGRL3
212826_s_at	0.001058082	0	0	0	SLC25A6
223044_at	0	4.72E-06	0	0	SLC40A1
201664_at	-0.00281327	0	0	0	SMC4
204466_s_at	0.000431789	0	0	0	SNCA
212560_at	0	0	0	-0.000796242	SORL1
201858_s_at	0	0.000305885	0	0	SRGN
224700_at	0	0	0	-0.000458156	STT3B
223939_at	-0.000372006	0	0.000186306	0	SUCNR1
216920_s_at	0	0	0.000633307	0	TARP
217733_s_at	0	0	0.001293331	0	TMSB10
224836_at	0	0	0	0.001784012	TP53INP2
215382_x_at	0.001024349	0	0	0	TPSAB1
208763_s_at	0.000888126	0	0	0	TSC22D3
209118_s_at	0	0	0	-0.000936702	TUBA1A
201008_s_at	-6.02E-05	0	0	0	TXNIP
201009_s_at	0	0.000719636	0	0	TXNIP
202589_at	-0.000553556	0	0	0	TYMS
208997_s_at	0.000204044	0	0	0	UCP2
204620_s_at	-0.00133927	0	0	0	VCAN
215646_s_at	-0.000212932	0	0	0	VCAN
221731_x_at	0	0	0	1.05E-05	VCAN
200670_at	-0.001830076	0	0	0	XBP1
227671_at	0	-0.000106502	0	0	XIST
213655_at	0	0	0	0.001603208	YWHAE
217741_s_at	0	0	0	0.0014386	ZFAND5
201531_at	0.000447636	0	0	0	ZFP36
201368_at	0	-0.000660923	0	0.000978681	ZFP36L2

S5 Differentially expressed genes for the inverse 16 subtype

Top 20 up- and down-regulated genes for the inverse 16 samples compared to all other subtypes.

Fold Change	ID	Gene Symbol
10.663	206135_at	ST18
8.649	201497_x_at	MYH11
6.303	212358_at	CLIP3
5.743	206682_at	CLEC10A
5.609	204885_s_at	MSLN
5.373	207961_x_at	MYH11
5.365	204787_at	VSIG4
5.206	1556034_s_at	MTMR11
5.068	241525_at	LOC200772
5.016	222862_s_at	AK5
4.859	205076_s_at	MTMR11
4.798	200665_s_at	SPARC
4.666	1564796_at	EMP1
4.479	205330_at	MN1
4.375	201506_at	TGFBI
4.336	212298_at	NRP1
4.274	224724_at	SULF2
4.233	233555_s_at	SULF2
4.17	203060_s_at	PAPSS2
4.166	201324_at	EMP1
4.032	232523_at	MEGF10
-5.934	212070_at	GPR56
-5.967	201670_s_at	MARCKS (includes EG:4082)
-6.051	217963_s_at	NGFRAP1
-6.121	211748_x_at	PTGDS
-6.213	214183_s_at	TKTL1
-6.233	229638_at	IRX3
-6.387	205801_s_at	RASGRP3
-6.403	201669_s_at	MARCKS (includes EG:4082)
-6.546	219218_at	BAHCC1
-7.125	213110_s_at	COL4A5
-7.227	206940_s_at	POU4F1
-7.383	211031_s_at	CLIP2

-7.428	211341_at	POU4F1
-7.598	201427_s_at	SEPP1
-7.632	235521_at	HOXA3 (includes EG:3200)
-9.109	206390_x_at	PF4
-9.616	213844_at	HOXA5
-10.534	214651_s_at	HOXA9
-11.729	209905_at	HOXA9
-11.755	223044_at	SLC40A1
-14.165	214146_s_at	PPBP

S6 Lasso: Prediction Table

		Test set error		Sensitivity	Specificity	Predictive Value	
		Neg	Pos	%	%	Neg	Pos
Case 1							
	Other	6/81	4/180	98	93	95	97
	t(15;17)	0/243	0/18	100	100	100	100
	t(8;21)	1/239	0/22	100	100	100	96
	inv(16)	2/238	0/23	100	99	100	92
	CEBPa	1/243	6/18	67	100	97	92
Case 2							
	Other	18/119	13/160	92	85	89	89
	NPM1+/FLT3ITD-	18/237	12/32	63	92	95	53
	NPM1-/FLT3ITD+	14/236	18/33	45	94	92	52
	NPM1+/FLT3ITD+	12/226	19/43	56	95	92	67

The following calculations were used for evaluation measures: sensitivity=true positives/(true positive + false negatives), specificity=true negatives/(true negatives + false positives), positive predictive value=true positives/(true positives + false positives), negative predictive value=true negatives /(true negatives + false negatives)

Work document

Multinomial logistic regression

Introduction:

The interest in statistical classification for critical applications such as diagnoses of patient samples based on supervised learning is rapidly growing. The primal interest is to design classifiers for different forms of decision support in performance sensitive applications, e.g. biomedicine. Important examples are predictions of tumor subtype and clinical outcome based on mRNA levels in tumor samples using modern large-scale microarray technologies.

Multinomial logistic regression:

One of these supervised learning algorithms is the multinomial logistic regression, which is based upon the dichotomous logistic regression principle. In statistics it is used for the prediction of the probability of occurrence of an event by fitting data to a logistic curve. In a two-class problem we can state the following:

- If a sample belongs to a specific class y_i with probability p_i, it has odds $\frac{p_i}{1-p_i}$.
- The vector $\vec{x}_i$ consists out of the data for sample i, such as a microarray expression profile.
- Odds have the range $[0, \infty)$.

We also make some additional assumption:

- The response y_i is Bernoulli distributed, such that: $p(y_i = 1|\vec{x}_i) = p$.
- The dichotomous logistic regression principle is described by a linear predictor: $\ln\left(\frac{p_i}{1-p_i}\right) = \eta_i$.
- The linear predictor is described by: $\eta_i = \beta_0 + \vec{x}_i^t \vec{\beta}$.
- Under conventional notation:

$$\eta_i = [1 \quad \vec{x}_i] \begin{bmatrix} \beta_0 \\ \vec{\beta} \end{bmatrix}$$

Using the definitions and assumptions stated above we can define the equations for the dichotomous logistic regression. Given the fact that:

$$p(y_i = 0|\vec{x}) + p(y_i = 1|\vec{x}) = 1$$

We can show, for the logistic regression equation of $p(y_i = 0|\vec{x})$, that:

$$\ln\left(\frac{p(y_i = 1|\vec{x})}{1 - p(y_i = 1|\vec{x})}\right) = \ln\left(\frac{p(y_i = 1|\vec{x})}{p(y_i = 0|\vec{x})}\right) = \ln\left(\frac{1 - p(y_i = 0|\vec{x})}{p(y_i = 0|\vec{x})}\right) = \vec{x}_i^t \vec{\beta}$$

$$\frac{1}{p(y_i = 0|\vec{x})} = e^{\vec{x}_i^t \vec{\beta}} + 1$$

$$p(y_i = 0|\vec{x}) = \frac{1}{e^{\vec{x}_i^t\vec{\beta}} + 1}$$

For the case $p(y_i = 1|\vec{x})$, this is easily generalized to:

$$p(y_i = 1|\vec{x}) = 1 - \ p(y_i = 0|\vec{x}) = 1 - \frac{1}{e^{\vec{x}_i^t\vec{\beta}} + 1} = \frac{e^{\vec{x}_i^t\vec{\beta}}}{e^{\vec{x}_i^t\vec{\beta}} + 1}$$

The dichotomous logistic regression model is easily constructed, but the polytomous logistic regression model becomes more involved. The multinomial logistic regression model was first introduced by McFadden (1974) for outcomes for more than two levels. The measurement scale for this model should be known in advance as there are the different types of models, namely the nominal scaled outcome variable and the ordinal scaled variable. As of now we are focusing only on the nominal scaled outcome variable. We shall show the construction of the generalized logistic model for three outcome categories. The construction of the model for more outcome categories should be self-evident for the reader.

Let us assume that the categories of the outcome variable y_i are coded as 0, 1 or 2. Recall that the logistic regression model for a binary outcome variable was parameterized in terms of the logit of $y_i = 1$ versus $y_i = 0$. In the three category model we have two logit functions: one for $y_i = 1$ versus $y_i = 0$, the other for $y_i = 2$ versus $y_i = 0$. In this case we taken the group coded as $y_i = 0$ as reference outcome value. The logit for comparing $y_i = 2$ to $y_i = 1$ may be obtained as the difference between the logit of $y_i = 2$ versus $y_i = 1$ and the logit of $y_i = 1$ versus $y_i = 0$. We will denote the two logit functions as:

$$\ln\left(\frac{p(y_i = 1|\vec{x}_i^t)}{p(y_i = 0|\vec{x}_i^t)}\right) = \vec{x}_i^t\vec{\beta}_1$$
$$\ln\left(\frac{p(y_i = 2|\vec{x}_i^t)}{p(y_i = 0|\vec{x}_i^t)}\right) = \vec{x}_i^t\vec{\beta}_2$$

In this case we have taken the group $y_i = 0$ as reference category and will be a topic of discussion later in this document. Using the definition:

$$p\left(y_i = 0\middle|\vec{x}_i^t\right) + p\left(y_i = 1\middle|\vec{x}_i^t\right) + p\left(y_i = 2\middle|\vec{x}_i^t\right) = 1$$

We state:

$$\frac{p(y_i = 1|\vec{x}_i^t)}{p(y_i = 0|\vec{x}_i^t)} = e^{\vec{x}_i^t\vec{\beta}_1}$$
$$\frac{p(y_i = 2|\vec{x}_i^t)}{p(y_i = 0|\vec{x}_i^t)} = e^{\vec{x}_i^t\vec{\beta}_2}$$

We can also show that:

$$\ln\left(\frac{p\left(y_i = 2\middle|\vec{x}_i^t\right)}{p\left(y_i = 1\middle|\vec{x}_i^t\right)}\right) = \ln\left(p\left(y_i = 2\middle|\vec{x}_i^t\right)\right) - \ln\left(p\left(y_i = 1\middle|\vec{x}_i^t\right)\right)$$

$$= \ln\left(\frac{p(y_i = 2|\vec{x}_i^t)p(y_i = 0|\vec{x}_i^t)}{p(y_i = 0|\vec{x}_i^t)}\right) - \ln\left(\frac{p(y_i = 1|\vec{x}_i^t)p(y_i = 0|\vec{x}_i^t)}{p(y_i = 0|\vec{x}_i^t)}\right)$$

$$= \ln\left(\frac{p(y_i = 2|\vec{x}_i^t)}{p(y_i = 0|\vec{x}_i^t)}\right) + \ln\left(p(y_i = 0|\vec{x}_i^t)\right) - \ln\left(\frac{p(y_i = 1|\vec{x}_i^t)}{p(y_i = 0|\vec{x}_i^t)}\right) - \ln(p(y_i = 0|\vec{x}_i^t))$$

$$= \ln\left(\frac{p(y_i = 2|\vec{x}_i^t)}{p(y_i = 0|\vec{x}_i^t)}\right) - \ln\left(\frac{p(y_i = 1|\vec{x}_i^t)}{p(y_i = 0|\vec{x}_i^t)}\right) = \vec{x}_i^t(\vec{\beta}_2 - \vec{\beta}_1)$$

Thus:

$$\frac{p(y_i = 2|\vec{x}_i^t)}{p(y_i = 1|\vec{x}_i^t)} = e^{\vec{x}_i^t(\vec{\beta}_2 - \vec{\beta}_1)}$$

Having defined all preliminary equations we can now derive the logit functions of each individual outcome category:

$$\frac{p(y_i = 2|\vec{x}_i^t)}{p(y_i = 1|\vec{x}_i^t)} = \frac{1 - p(y_i = 1|\vec{x}_i^t) - p(y_i = 0|\vec{x}_i^t)}{p(y_i = 1|\vec{x}_i^t)} = \frac{1}{p(y_i = 1|\vec{x}_i^t)} - 1 - \frac{1}{e^{\vec{x}_i^t\vec{\beta}_1}} = e^{\vec{x}_i^t(\vec{\beta}_2 - \vec{\beta}_1)}$$

$$\frac{1}{p(y_i = 1|\vec{x}_i^t)} = 1 + \frac{1}{e^{\vec{x}_i^t\vec{\beta}_1}} + e^{\vec{x}_i^t(\vec{\beta}_2 - \vec{\beta}_1)} = \frac{1 + e^{\vec{x}_i^t\vec{\beta}_1} + e^{\vec{x}_i^t\vec{\beta}_2}}{e^{\vec{x}_i^t\vec{\beta}_1}}$$

$$p(y_i = 1|\vec{x}_i^t) = \frac{e^{\vec{x}_i^t\vec{\beta}_1}}{1 + e^{\vec{x}_i^t\vec{\beta}_1} + e^{\vec{x}_i^t\vec{\beta}_2}}$$

For $p(y_i = 0|\vec{x}_i^t)$:

$$\frac{p(y_i = 1|\vec{x}_i^t)}{p(y_i = 0|\vec{x}_i^t)} = e^{\vec{x}_i^t\vec{\beta}_1}$$

$$\frac{1}{p(y_i = 0|\vec{x}_i^t)} = \frac{e^{\vec{x}_i^t\vec{\beta}_1}}{\frac{e^{\vec{x}_i^t\vec{\beta}_1}}{1 + e^{\vec{x}_i^t\vec{\beta}_1} + e^{\vec{x}_i^t\vec{\beta}_2}}} = 1 + e^{\vec{x}_i^t\vec{\beta}_1} + e^{\vec{x}_i^t\vec{\beta}_2}$$

$$p(y_i = 0|\vec{x}_i^t) = \frac{1}{1 + e^{\vec{x}_i^t\vec{\beta}_1} + e^{\vec{x}_i^t\vec{\beta}_2}}$$

Finally for $p(y_i = 2|\vec{x}_i^t)$:

$$p(y_i = 2|\vec{x}_i^t) = 1 - p(y_i = 1|\vec{x}_i^t) - p(y_i = 0|\vec{x}_i^t) = 1 - \frac{e^{\vec{x}_i^t\vec{\beta}_1}}{1 + e^{\vec{x}_i^t\vec{\beta}_1} + e^{\vec{x}_i^t\vec{\beta}_2}} - \frac{1}{1 + e^{\vec{x}_i^t\vec{\beta}_1} + e^{\vec{x}_i^t\vec{\beta}_2}}$$

$$p(y_i = 2|\vec{x}_i^t) = \frac{e^{\vec{x}_i^t\vec{\beta}_2}}{1 + e^{\vec{x}_i^t\vec{\beta}_1} + e^{\vec{x}_i^t\vec{\beta}_2}}$$

Hence we have found the logit functions for all outcome categories when using the group $y_i = 0$ as reference category. As noted earlier on we have constructed a multinomial logistic regression model with a reference category. The additional advantage of this type of modeling is that the model is not

overparameterized. We show this later on, but intuitively it indicates that if we know the regression coefficient vectors $\vec{\beta}_1, \cdots, \vec{\beta}_{g-1}$ we need not to know regression coefficient vector $\vec{\beta}_g$, as it is already pre-determined due to the other regression coefficient vectors. For convenient notation and reasons that become clearer later in this document we define the logit functions for each category outcome variable as:

$$\mu_{is} = p(y_i = s|\vec{x}_i) = \frac{e^{\eta_{is}}}{\sum_{t=1}^{g} e^{\eta_{it}}}$$

In this research we are trying to classify tumor samples originating from multiple different classes. Where the class labels are defined as:

$$y_{is} = \mathbf{1}_{\{y_i=s\}} \quad for\ i = 1, \cdots, n\ and\ s = 1, \cdots, g$$
$$i: Sample\ index$$
$$s: Class\ label$$

Where are dealing here with a g-class classification problem of n samples. We have also the covariates of the regression, which normally are the data vectors of the samples. We define them as $\vec{X}_0, \vec{X}_1, \cdots, \vec{X}_p$, where each vector contains data of a specific feature for all samples (such as the expression measurement of a specific gene for each samples), with exception of $\vec{X}_0$ which consists only out of elements equal to 1 such that the offset of the regression is conveniently integrated in the regression. This can be written in matrix form as:

$$\boldsymbol{X} = \begin{bmatrix} \vec{X}_0 & \vec{X}_1 & \cdots & \vec{X}_p \end{bmatrix}$$

We are thus dealing with a g-class classification problem with p features. Due to the primal fact that we are dealing with a multi-class problem we also have a regression coefficient vector for each class, defined as $\vec{\beta}_i, i = 1, \cdots, g$. Using this notation we can define the linear predictors as:

$$\eta_{is} = \vec{x}_i^t \vec{\beta}_s$$

or in matrix form:

$$\eta_{is} = \sum_{k=1}^{p+1} X_{ik}\beta_{ks}$$

In this formulation of the model we have a regression coefficient β_{ks} for each combination of covariate k and outcome category s. Suppose we have outcomes $Y_1, \cdots, Y_n$, a corresponding $n \times (p+1)$ data matrix of covariates X and make the simplifying substitution $\mu_{is} = p(Y_i = s|\vec{x}_i)$. For notational convenience we write the y_{is}, μ_{is} and η_{is} in the form of long $ng \times 1$ vectors:
$\boldsymbol{y} = (y_{11}, \cdots, y_{n1}, \cdots, y_{1g}, \cdots, y_{ng})^T$, $\boldsymbol{\mu} = (\mu_{11}, \cdots, \mu_{n1}, \cdots, \mu_{1g}, \cdots, \mu_{ng})^T$ and
$\boldsymbol{\eta} = (\eta_{11}, \cdots, \eta_{n1}, \cdots, \eta_{1g}, \cdots, \eta_{ng})^T$. The linear predictors $\boldsymbol{\eta}$ are related to the vector of parameters $\boldsymbol{\beta} = (\beta_{01}, \beta_{11}, \cdots, \beta_{p1}, \cdots, \beta_{0g}, \cdots, \beta_{pg})^T$ through $\boldsymbol{\eta} = \boldsymbol{X\beta}$, where $\boldsymbol{X} = I_g \otimes X$,where $\otimes$ is the Kronecker product and I_g the $g \times g$ identity matrix.

Fitting the model:

There is no analytical way to fit the model to the observed data, as we do not have a closed expression. To solve this problem we are going to make use of the likelihood function of the model. Recall that the probabilities of the category outcomes are Bernoulli distributed. If random variables are independently distributed we can state that:

$$L(\theta) = p(y_1, \cdots, y_n|\theta) = p(y_1|\theta) \cdots p(y_n|\theta) = \prod_{\boldsymbol{i=1}}^{\boldsymbol{n}} f_\theta(x_i)$$

Hence, we are trying to maximize the conditional probability of the observed data given the parameters. We are thus trying to find the parameters which best explains the outcomes given the model structure. In case of Bernoulli variables and our logistic regression principle we can state:

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \prod_{s=1}^{g} \mu_{is}^{y_{is}}$$

To maximize this expression we are going to make use of the log-likelihood function which has its maxima at exactly the same parameter values.

$$l(\boldsymbol{\beta}) = \ln(L(\boldsymbol{\beta})) = \sum_{i=1}^{n} \sum_{s=1}^{g} y_{is} \ln(\mu_{\mathrm{is}}) = \sum_{i=1}^{n} \sum_{s=1}^{g} y_{is} \ln\left(\frac{e^{\eta_{is}}}{\sum_{t=1}^{g} e^{\eta_{it}}}\right) = \sum_{i=1}^{n} \sum_{s=1}^{g} y_{is}(\eta_{is} - \ln(\sum_{t=1}^{g} e^{\eta_{it}}))$$

To find the gradient of this log-likelihood function with respect to the beta coefficients we can show the procedure without loss of generality for one partial derivative.

$$\frac{\partial}{\partial \beta_{kh}} l(\boldsymbol{\beta}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \frac{\partial}{\partial \beta_{kh}} \sum_{s=1}^{g} y_{is} \left(\eta_{is} - \ln\left(\sum_{t=1}^{g} e^{\eta_{it}}\right)\right) = \sum_{i=1}^{n} \boldsymbol{X}_{ik}(y_{ih} - \mu_{ih})$$

This general procedure can be applied to each covariate *k* for each class *s*. To write this procedure in efficient matrix operations we constructed the new design matrix by the Kronecker product resulting in a block matrix and augmented the indicator functions for each class and predicted probabilities to their respective vectors $\boldsymbol{y}$ and $\boldsymbol{\mu}$. In terms of block matrices this would look like:

$$\boldsymbol{X}^* = \begin{bmatrix} \boldsymbol{X} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X} & & \vdots \\ \vdots & & \ddots & \boldsymbol{0} \\ \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{X} \end{bmatrix} = I_g \otimes \boldsymbol{X}, \boldsymbol{y} = \begin{bmatrix} \vec{y}_1 \\ \vec{y}_2 \\ \vdots \\ \vec{y}_n \end{bmatrix}, \boldsymbol{\mu} = \begin{bmatrix} \vec{\mu}_1 \\ \vec{\mu}_2 \\ \vdots \\ \vec{\mu}_n \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \vec{\beta}_1 \\ \vec{\beta}_2 \\ \vdots \\ \vec{\beta}_g \end{bmatrix}$$

Using these definitions we can define the gradient vector as:

$$\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = (\boldsymbol{X}^*)^T(\boldsymbol{y} - \boldsymbol{\mu})$$

For the Hessian we have to define two different cases that can arise:

Case one: The first first-order partial derivative is not equal to the second first-order partial derivative

$$\frac{\partial}{\partial \beta_{ks}} \sum_{i=1}^{n} \boldsymbol{X}_{ik}(y_{ih} - \mu_{ih}) = -\sum_{i=1}^{n} \boldsymbol{X}_{ik} \frac{\partial}{\partial \beta_{ks}} \mu_{ih} = -\sum_{i=1}^{n} \boldsymbol{X}_{ik} e^{\eta_{ih}} \frac{\partial}{\partial \beta_{ks}} \frac{1}{\sum_{t=1}^{g} e^{\eta_{it}}}$$
$$= \sum_{i=1}^{n} \boldsymbol{X}_{ik}^2 \, e^{\eta_{ih}} e^{\eta_{is}} \frac{1}{\left(\sum_{t=1}^{g} e^{\eta_{it}}\right)^2} = \sum_{i=1}^{n} \boldsymbol{X}_{ik}^2 \mu_{ih} \mu_{is}$$

Case two: The first first-order partial derivative is equal to the second first-order partial derivative

$$\frac{\partial}{\partial \beta_{kh}} \sum_{i=1}^{n} \boldsymbol{X}_{ik}(y_{ih} - \mu_{ih}) = -\sum_{i=1}^{n} \boldsymbol{X}_{ik} \frac{\partial}{\partial \beta_{ks}} \mu_{ih} = -\sum_{i=1}^{n} \boldsymbol{X}_{ik} \frac{\partial}{\partial \beta_{ks}} \frac{e^{\eta_{ih}}}{\sum_{t=1}^{g} e^{\eta_{it}}}$$
$$= -\sum_{i=1}^{n} \boldsymbol{X}_{ik} \frac{\boldsymbol{X}_{ik}\left(e^{\eta_{ih}} \sum_{t=1}^{g} e^{\eta_{it}}\right) - \boldsymbol{X}_{ik}(e^{\eta_{ih}})^{\boldsymbol{2}}}{\sum_{t=1}^{g} e^{\eta_{it}}} = -\sum_{i=1}^{n} \boldsymbol{X}_{ik}^2 \frac{e^{\eta_{ih}}\left(\sum_{t=1}^{g} e^{\eta_{it}} - e^{\eta_{ih}}\right)}{\left(\sum_{t=1}^{g} e^{\eta_{it}}\right)^2}$$
$$= -\sum_{i=1}^{n} \boldsymbol{X}_{ik}^2 \, \mu_{ih}(1 - \mu_{ih})$$

We now know the second order partial derivatives for all cases and we can construct the Hessian conveniently by matrix operations. The Hessian is defined as:

$$\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^t} = -(\boldsymbol{X}^*)^T \boldsymbol{Z} \boldsymbol{X}^*$$

Where the $ng \times ng$ matrix $\boldsymbol{Z}$ is given by:

$$\boldsymbol{Z} = \begin{pmatrix} Z^{11} & Z^{12} & \cdots & Z^{1g} \\ Z^{21} & Z^{22} & & \vdots \\ \vdots & & \ddots & \\ Z^{g1} & \cdots & & Z^{gg} \end{pmatrix}$$

Where each W^{ij} is a diagonal matrix with entries:

$$diag(Z^{st}) = diag(Z^{ts}) = \begin{cases} (-\mu_{1s}\mu_{1t}, \cdots, -\mu_{ns}\mu_{nt})^T & if\ s \neq t \\ (\mu_{1s}(1 - \mu_{1s}), \cdots, \mu_{ns}(1 - \mu_{ns}))^T & if\ s = t \end{cases}$$

Concavity:

An additional advantage of this log-likelihood function is its concavity, although it is not strict concave. We can show this by proofing the assumption that the Hessian is semi-definite negative. To do this we will use the most common factorization of non-square matrices called the singular value decomposition. We assume the reader to be well educated in this subject.

For any $m \times n$ matrix A, the $n \times n$ matrix $A^T A$ is symmetric and hence can be orthogonally diagonalized, by the Spectral Theorem. Not only are the eigenvalues of $A^T A$ an element of the set $\mathbb{R}$, they are also non-negative. To show this, let λ be an eigenvalue of $A^T A$ wit corresponding eigenvector $\vec{v}$. We can show that

$$0 \leq \|A\vec{v}\|_2^2 = (A\vec{v})^T(A\vec{v}) = \vec{v}^T A^T A\vec{v} = \vec{v}^T\lambda\vec{v} = \lambda\vec{v}^T\vec{v} = \lambda\|\vec{v}\|_2^2 = \lambda$$

Recall that the eigenvectors of the symmetric matrix $A^T A$ is an orthonormal basis for the column space of this matrix, hence its norm l^2 is equal to 1. We can now state:

$$-(\boldsymbol{X}^*)^T\boldsymbol{Z}\boldsymbol{X}^* = -(\boldsymbol{X}^*)^T\boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^T\boldsymbol{X}^* = -(\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{Q}^T\boldsymbol{X}^*)^T\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{Q}^T\boldsymbol{X}^*$$

Making the substitution $\boldsymbol{B} = \boldsymbol{D}^{\frac{1}{2}}\boldsymbol{Q}^T\boldsymbol{X}^*$, we get:

$$-(\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{Q}^T\boldsymbol{X}^*)^T\boldsymbol{D}^{\frac{1}{2}}\boldsymbol{Q}^T\boldsymbol{X}^* = -\boldsymbol{B}^T\boldsymbol{B}$$

It shows that the Hessian can be decomposed into two transposed matrices and proofs the fact that the eigenvalues of the Hessian are all negative semi-definite (due to the negative sign in front of the formula). This indicates that the log-likelihood function is concave and has major advantages when using optimization procedures such as Newton optimization to find the appropriate regression coefficients. When optimizing concave functions the local optimum is equal to the global optimum.

Optimization:

In mathematics, Newton's method is a well-known algorithm for finding roots of equations in one or more dimensions. It can also be used to find local maxima and local minima of functions by noticing that if a real number x^* is stationary point of a function $f(x)$, then x^* is a root of the derivative $f(x)$, and therefore one can solve for x^* by applying Newton's method to $f(x)$. First we should attain the second order Taylor expansion of the function $f(x)$. This is given by:

$$f(x + \Delta x) = f(x) + f'(x)\Delta x + \frac{1}{2}f''(x)\Delta x^2$$

And attains its extremum when Δx solves the linear equation:

$$f'(x) + f''(x)\Delta x = 0$$

And $f''(x)$ is positive. Thus, provided that $f(x)$ is a twice-differential function and the initial guess x_0 is chosen close enough to x^*, the sequence $\{x_n\}$ defined by:

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}, n \geq 0$$

Will converge towards x^*. This iterative scheme can be generalized to several dimensions by replacing the derivative with the gradient, $\nabla f(\vec{x})$, and the reciprocal of the second derivative with the inverse of the Hessian matrix $H(\vec{x})$. One obtains the iterative scheme:

$$\vec{x}_{n+1} = \vec{x}_n - H^{-1}(\vec{x}_n)\nabla f(\vec{x}_n), n \geq 0$$

The advantage that the log-likelihood function is concave certifies that the algorithm shall converge to the true maximum. As shown earlier, we already have the gradient and Hessian.

Computational problems:

One of the problems when fitting the multinomial logistic regression model, by maximizing the log-likelihood function, is that the Hessian is not invertible. The Hessian is a $pg \times pg$ matrix that is singular, due to the fact that the rank of this matrix is $p(g-1)$. This is solely due the fact that we have an overparameterized model, because we have not defined any reference category thus leading to dependence. To avoid this problem we could use the Moore-Penroose inverse of the Hessian, leading to the minimum length solution for $\boldsymbol{\beta}$.

$$\boldsymbol{\beta}_{n+1} = \boldsymbol{\beta}_n + ((\boldsymbol{X}^*)^T \boldsymbol{Z} \boldsymbol{X}^*)^+ (\boldsymbol{X}^*)^T (\boldsymbol{y} - \boldsymbol{\mu})$$

Theorem 1: Let $A = U\Sigma V^T$ be an Singular Value Decomposition for an $m \times n$ matrix A, where $\Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$ and D is an $r \times r$ diagonal matrix containing the nonzero singular values $\sigma_1 \geq \cdots \geq \sigma_r > 0$ Of A. The Moore-Penrose inverse of A is the $n \times m$ matrix A^+, defined by:

$$A^+ = U\Sigma^+ V^T$$

Where Σ^+ is the $n \times m$ matrix

$$\Sigma^+ = \begin{bmatrix} D^{-1} & 0 \\ 0 & 0 \end{bmatrix}$$

The Moore-Penrose inverse is normally used when the matrix A has linear dependent columns. In this case $A^T A$ is not invertible. This leads to infinitely many solutions when one is using the pseudo-inverse $(A^T A)^{-1} A^T$). With the Moore-Penrose matrix we are given the solution $\bar{x}$ of minimum length (i.e. the one closest to the origin).

Proof: Let A be an $m \times n$ matrix of rank r with Singular value Decomposition $A = U\Sigma V^T$. Let $\vec{y} = V^T \vec{x}$ and let $\vec{c} = U^T \vec{b}$, we write $\vec{y}$ and $\vec{c}$ in block form:

$$\vec{y} = \begin{bmatrix} \vec{y}_1 \\ \vec{y}_2 \end{bmatrix}, \quad \vec{c} = \begin{bmatrix} \vec{c}_1 \\ \vec{c}_2 \end{bmatrix} \quad where\ \vec{y}_1 \wedge \vec{c}_1 \in \mathbb{R}^r$$

We wish to minimize $\left\|\vec{b} - A\vec{x}\right\|_2$, or equivalently, $\left\|\vec{b} - A\vec{x}\right\|_2^2$. We use the fact that U^T is orthogonal.

$$\left\|\vec{b} - A\vec{x}\right\|_2^2 = \left\|U^T(\vec{b} - A\vec{x})\right\|_2^2 = \left\|U^T(\vec{b} - U\Sigma V^T\vec{x})\right\|_2^2 = \|\vec{c} - U^T U\Sigma V^T\vec{x}\|_2^2$$
$$\|\vec{c} - \Sigma\vec{y}\|_2^2 = \left\|\begin{bmatrix}\vec{c}_1\\ \vec{c}_2\end{bmatrix} - \begin{bmatrix}D & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\vec{y}_1\\ \vec{y}_2\end{bmatrix}\right\|_2^2 = \left\|\begin{bmatrix}\vec{c}_1 - D\vec{y}_1\\ \vec{c}_2\end{bmatrix}\right\|_2^2$$

The only part of this expression that we have any control over is $\vec{y}_1$, so the minimum norm occurs when $\vec{c}_1 - D\vec{y}_1$, or equivalently when $\vec{y}_1 = D^{-1}\vec{c}_1$. So all least squares solutions $\vec{x}$ are of the form:

$$\vec{x} = V\vec{y} = V\begin{bmatrix}D^{-1}\vec{c}_1\\ \vec{y}_2\end{bmatrix}$$

Set as the minimum length solution:

$$\bar{x} = V\bar{y} = V\begin{bmatrix}D^{-1}\vec{c}_1\\ \vec{0}\end{bmatrix}$$

This is the least squares solution with minimum length. To show this, let's suppose that:

$$\vec{x}' = V\vec{y}' = V\begin{bmatrix}D^{-1}\vec{c}_1\\ \vec{y}_2\end{bmatrix}, where\ \vec{y}_2 \neq \vec{0}$$

We can show that:

$$\|\bar{x}\|_2 = \|V\bar{y}\|_2 = \|\bar{y}\|_2 < \|\vec{y}'\|_2 = \|V\vec{y}'\|_2 = \|\vec{x}'\|_2$$

Finally we show that $\bar{x}$ is equal to $A^+\vec{b}$:

$$\bar{x} = V\bar{y} = V\begin{bmatrix}D^{-1}\vec{c}_1\\ \vec{0}\end{bmatrix} = V\begin{bmatrix}D & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\vec{c}_1\\ \vec{c}_2\end{bmatrix} = V\Sigma^+\vec{c} = V\Sigma^+U^T\vec{b} = A^+\vec{b}$$

Hence we have found the least squares solution with minimal length.

<u>Alternative view:</u>

One of the major problems of the overparameterization is the invariance of the model in certain directions. Let us yet again define the long vector $\boldsymbol{\beta}$**:**

$$\boldsymbol{\beta} = \begin{bmatrix}\vec{\beta}_1\\ \vec{\beta}_2\\ \vdots\\ \vec{\beta}_g\end{bmatrix}$$

If we were to add a constant vector to each regression coefficient vector:

$$\boldsymbol{\beta} + \boldsymbol{c} = \begin{bmatrix} \vec{\beta}_1 \\ \vec{\beta}_2 \\ \vdots \\ \vec{\beta}_g \end{bmatrix} + \begin{bmatrix} \vec{c} \\ \vec{c} \\ \vdots \\ \vec{c} \end{bmatrix}$$

We can easily show that:

$$\mu_{is} = p(y_i = s|\vec{x}_i) = \frac{e^{\vec{x}_i^t\left(\vec{\beta}_s + \vec{c}\right)}}{\sum_{t=1}^{g} e^{\vec{x}_i^t\left(\vec{\beta}_t + \vec{c}\right)}} = \frac{e^{\vec{x}_i^t \vec{c}}}{e^{\vec{x}_i^t \vec{c}}} \frac{e^{\vec{x}_i^t \vec{\beta}_s}}{\sum_{t=1}^{g} e^{\vec{x}_i^t \vec{\beta}_t}} = \frac{e^{\vec{x}_i^t \vec{\beta}_s}}{\sum_{t=1}^{g} e^{\vec{x}_i^t \vec{\beta}_t}}$$

Hence, the model is invariant to the translation of the regression coefficient vectors in one particular direction. To avoid the overparameterization we can also define the following condition:

$$\hat{\beta}_{k1} + \hat{\beta}_{k2} + \cdots + \hat{\beta}_{kg} = 0 \; for \; k = 1, \cdots, p$$

Or when writing in matrix form, we can sum over the rows:

$$\boldsymbol{\beta}^* = \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1g} \\ \beta_{21} & \beta_{22} & & \beta_{2g} \\ \vdots & & \ddots & \vdots \\ \beta_{p1} & \beta_{p2} & \cdots & \beta_{pg} \end{bmatrix}$$

We are trying to find a vector $\hat{\beta}$, such that:

$$V^T \boldsymbol{\beta} = \vec{0}$$

Where V^T is:

$$V^T = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 & & 1 & 0 & & & 0 \\ 0 & 1 & 0 & \cdots & 0 & \cdots & 0 & 1 & 0 & & 0 \\ \vdots & & & \ddots & \vdots & & \vdots & & & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 & & 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

Or equivalently

$$V^T = \begin{bmatrix} {I_p}^1 & {I_p}^2 & \cdots & {I_p}^g \end{bmatrix}$$

The last formulation indicates that the matrix V^T consists out of g block matrices which are identically to the p-dimensional identity matrix I_p. From theory we know that this vector $\hat{\beta}$ is in the null space of V^T and thus is orthogonal to the column space of its transpose V, as shown in Figure 1.

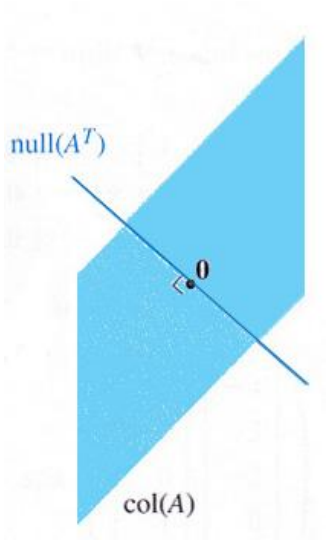


Figure 1: Orthogonal spaces

An additional advantage is that the column vectors of the matrix V are linearly independent. Hence, we can use the pseudo-inverse to construct projection matrices. We state that the vector $\vec{\beta}$ can be orthogonally projected on the column space of V and the null space of V^T, as shown in Figure 2:

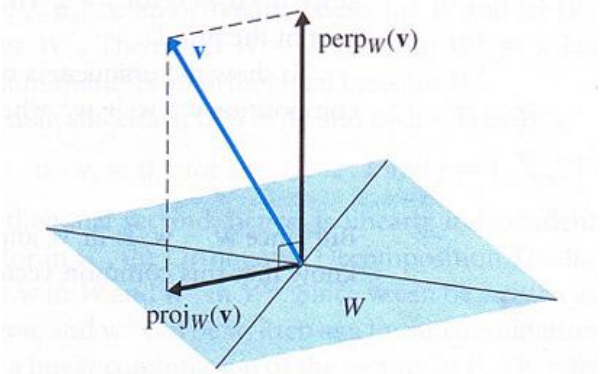


Figure 2: Orthogonal Decomposition of vector $\vec{v}$

Thus the vector $\vec{\beta}$ can be expressed as:

$$\vec{\beta} = proj_V(\vec{\beta}) + perp_V(\vec{\beta}) = V(V^TV)^{-1}V^T\vec{\beta} + (I - V(V^TV)^{-1}V^T)\vec{\beta}$$

We sometimes conveniently write the matrix $V(V^TV)^{-1}V^T$ as the matrix H (hat matrix), which has some nice properties:

Theorem**:**

a) The Hat matrix is symmetric
b) The Hat matrix is idempotent, meaning that the eigenvalues are 0 or 1. Which also constitutes that the Hat matrix is semi-positive definite
c) If $\vec{y}\ \epsilon\ null(X^T)$ it will be projected to the $\vec{0}$ vector
d) If $\vec{y}\epsilon\ col(X)$ it will remain $\vec{y}$ after projection
e) There is only a least square solution when the columns of X are linearly independent

Proof**:**

a) $H^T = (X(X^TX)^{-1}X^T)^T = X(X^TX)^{-1}X^T$
b) $HH = X(X^TX)^{-1}X^TX(X^TX)^{-1}X^T = X(X^TX)^{-1}X^T = H$
c) $H\vec{y} = X(X^TX)^{-1}X^T\vec{y} = X(X^TX)^{-1}\vec{0} = \vec{0}$
d) If $\vec{y}\epsilon\ col(X)$ it can be written as a linear combination such as: $\vec{y} = c_1\vec{x}_1 + \cdots + c_n\vec{x}_n = X\vec{c}$ this leads to $X(X^TX)^{-1}X^T\vec{y} = X(X^TX)^{-1}X^TX\vec{c} = X\vec{c} = \vec{y}$
e) If we go back to the form $X^TX\hat{\beta} = X^T\vec{y}$ we can prove that X^TX is invertible if the columns of X are linearly independent.

$$\hat{\beta}^TX^TX\hat{\beta} = \vec{0}$$
$$\left(\beta_0\vec{1} + \beta_1\overrightarrow{x_1} + \cdots + \beta_{p+1}\overrightarrow{x_p}\right)^T\left(\beta_0\vec{1} + \beta_1\overrightarrow{x_1} + \cdots + \beta_{p+1}\overrightarrow{x_p}\right) = \vec{0}\ if\ and\ only\ if$$
$$\hat{\beta} = \vec{0}\ if\ the\ columns\ of\ X\ are\ linearly\ independent$$

We are only interested in the solutions orthogonally projected on the null space of V^T. We can observe the following:

$$V = \begin{bmatrix} I_p \\ I_p \\ \vdots \\ I_p \end{bmatrix}$$

$$V^TV = [I_p \quad I_p \quad \cdots \quad I_p]\begin{bmatrix} I_p \\ I_p \\ \vdots \\ I_p \end{bmatrix} = gI_p$$

$$(V^TV)^{-1} = \frac{1}{g}I_p$$

$$V(V^TV)^{-1}V^T = \frac{1}{g}VV^T = \begin{bmatrix} I_p \\ I_p \\ \vdots \\ I_p \end{bmatrix}[I_p \quad I_p \quad \cdots \quad I_p] = \begin{bmatrix} \frac{1}{g}I_p^{11} & \cdots & \frac{1}{g}I_p^{1g} \\ \vdots & \ddots & \vdots \\ \frac{1}{g}I_p^{g1} & \cdots & \frac{1}{g}I_p^{gg} \end{bmatrix}$$

Finally the projection matrix on the null space of V^T is given by:

$$H_{null} = I_{pg} - V(V^TV)^{-1}V^T = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{1}{g}I_p^{11} & \cdots & \frac{1}{g}I_p^{1g} \\ \vdots & \ddots & \vdots \\ \frac{1}{g}I_p^{g1} & \cdots & \frac{1}{g}I_p^{gg} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{g-1}{g}I_p & -\frac{1}{g}I_P & \cdots & -\frac{1}{g}I_P \\ -\frac{1}{g}I_P & \frac{g-1}{g}I_p & & \vdots \\ \vdots & & \ddots & -\frac{1}{g}I_P \\ -\frac{1}{g}I_P & \cdots & -\frac{1}{g}I_P & \frac{g-1}{g}I_p \end{bmatrix}$$

If we look carefully at this matrix we can identify a pattern, namely:

$$H_{null}\boldsymbol{\beta} = \boldsymbol{\beta} - \overline{\boldsymbol{\beta}}$$

Where $\overline{\boldsymbol{\beta}}$ is the average of all the regression coefficients belonging to one class.

$$\overline{\boldsymbol{\beta}} = \begin{bmatrix} \overline{\boldsymbol{\beta}}_{\mathbf{1}} \\ \overline{\boldsymbol{\beta}}_{\mathbf{2}} \\ \vdots \\ \overline{\boldsymbol{\beta}}_{\boldsymbol{g}} \\ \overline{\boldsymbol{\beta}}_{\mathbf{1}} \\ \vdots \\ \vdots \\ \overline{\boldsymbol{\beta}}_{\boldsymbol{g}} \end{bmatrix}$$

We can easily show that for the condition states earlier this projection holds:

$$\left(\beta_{k1} - \bar{\beta}_k\right) + \left(\beta_{k2} - \bar{\beta}_k\right) + \cdots + \left(\beta_{kg} - \bar{\beta}_k\right), k = 1, \cdots, p$$

$$\sum_{j=1}^{g} \beta_{kg} - n\bar{\beta}_k = 0$$

As this project does nothing more than translate all regression coefficient vectors with the same amount and direction, the solution of the log-likelihood maximization stays the same but the overparameterization is resolved.

Rank and nullity projection matrix:

Additional properties of the projection matrix can be given. In this section we want to proof that the rank and nullity of the projection matrix is the same as that of the matrix V^T.

$$rank(V^T) + nullity(V^T) = p * g = rank(V(V^TV)^{-1}V^T) + nullity(V(V^TV)^{-1}V^T)$$

To show that $rank(V^T) = rank(V(V^TV)^{-1}V^T)$, it is enough to show that $nullity(V^T) = nullity(V(V^TV)^{-1}V^T)$. First of all we know that the $nullity(V^T) = p(g-1)$, because $rank(V^T) = p$. Let $\vec{x} \in null(V^T)$, such that $V^T\vec{x} = \vec{0}$. Then

$$V(V^TV)^{-1}V^T\vec{x} = V(V^TV)^{-1}\vec{0} = \vec{0}$$

And thus $\vec{x} \in null(V(V^TV)^{-1}V^T)$. Conversely, let $\vec{y} \in null(V(V^TV)^{-1}V^T)$, such that:

$$V(V^TV)^{-1}V^T\vec{y} = \vec{0}$$

This implies also that $\vec{y}^TV(V^TV)^{-1}V^T\vec{y} = \vec{0}$. But then:

$$\vec{y}^TV(V^TV)^{-1}V^T\vec{y} = \vec{y}^TVQD^{-1}Q^TV^T\vec{y} = \vec{y}^TVQD^{-\frac{1}{2}}D^{-\frac{1}{2}}Q^TV^T\vec{y}$$
$$= \left(D^{-\frac{1}{2}}Q^TV^T\vec{y}\right)^T\left(D^{-\frac{1}{2}}Q^TV^T\vec{y}\right) = \vec{0}$$

Remember that the orthogonal matrix is nothing more than a rotation matrix. A rotation matrix is one that preserves the norm (length) of the vector which can easily be shown by:

$$\|Q^T\vec{z}\|_2 = \sqrt{\vec{z}^TQQ^T\vec{z}} = \|\vec{z}\|_2$$

The matrix D is nothing more than a scaling matrix, shown by:

$$D^{-\frac{1}{2}}\vec{k} = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{\sqrt{\lambda_n}} \end{bmatrix}\begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix} = \begin{bmatrix} \frac{k_1}{\sqrt{\lambda_1}} \\ \frac{k_2}{\sqrt{\lambda_2}} \\ \vdots \\ \frac{k_n}{\sqrt{\lambda_n}} \end{bmatrix}$$

Hence

$$\left(D^{-\frac{1}{2}}Q^TV^T\vec{y}\right)^T\left(D^{-\frac{1}{2}}Q^TV^T\vec{y}\right) = \vec{0}$$

If and only if $V^T\vec{y} = \vec{0}$. Hence the rank and nullity are the same.

Basis:

It is a well-known fact that the eigenvectors of $V(V^TV)^{-1}V^T$ span the same space as the vectors belonging to $col(V)$. The orthogonal complements of these are the eigenvectors of $I - V(V^TV)^{-1}V^T$ or the vectors belonging to $null(V^T)$ and together they span the complete space with a dimensionality equal to the dimensionality of each vector. Using the orthogonal projection we can state that:

$$I - V(V^TV)^{-1}V^T = I - QDQ^T = (QQ^T - QDQ^T) = Q(I - D)Q^T$$

An additional property of the projection matrix is that it is idempotent implying that its eigenvalues are equal to zero or one, put mathematically:

$$\{D\}_{ij} = \begin{cases} 0 \vee 1 & i = j \\ 0 & i \neq j \end{cases}$$

The number of eigenvalues which are equal to 1 is in this case the same as the nullity of $V(V^TV)^{-1}V^T$, shown earlier to be $p(g-1)$. We can thus write

$$Q(I - D)Q^T = [\vec{q}_1 \quad \cdots \quad \vec{q}_{pg}] \begin{bmatrix} 1 & \cdots & 0 & \\ \vdots & \ddots & \vdots & \mathbf{0} \\ 0 & \cdots & 1 & \\ & \mathbf{0} & & \mathbf{0} \end{bmatrix} \begin{bmatrix} \vec{q}_1^T \\ \vdots \\ \vec{q}_{pg}^T \end{bmatrix} = \begin{bmatrix} \vec{q}_1 & \cdots & \vec{q}_{p(g-1)} \end{bmatrix} \begin{bmatrix} \vec{q}_1^T \\ \vdots \\ \vec{q}_{p(g-1)}^T \end{bmatrix}$$

We define:

$$\boldsymbol{W} = \begin{bmatrix} \vec{q}_1 & \cdots & \vec{q}_{p(g-1)} \end{bmatrix}$$

Hence we can write:

$$\widehat{\boldsymbol{\beta}} = (I - V(V^TV)^{-1}V^T)\boldsymbol{\beta} = \boldsymbol{W}\boldsymbol{W}^T\boldsymbol{\beta}$$

Here we have constructed the matrix $\boldsymbol{W}$, which is an semi-orthogonal matrix where its column vectors span the null space of V^T.

Optimization:

Now that we have reparameterized the model, we should also change the log-likelihood function in accordance. It is easy to show that:

$$\frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}} = \frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \widehat{\boldsymbol{\beta}}} \frac{\partial \widehat{\boldsymbol{\beta}}}{\partial \boldsymbol{\beta}} = \boldsymbol{W}\boldsymbol{W}^T \frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \widehat{\boldsymbol{\beta}}} = \boldsymbol{W}\boldsymbol{W}^T \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

The last term comes from the fact that both gradients are equal, because the log-likelihood function is translation invariant. Finally we can also write:

$$\widehat{\boldsymbol{\beta}} = \boldsymbol{W}\vec{\gamma}$$

Where the gradient of the log-likelihood function changes to:

$$\frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \vec{\gamma}} = \frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \widehat{\boldsymbol{\beta}}} \frac{\partial \widehat{\boldsymbol{\beta}}}{\partial \vec{\gamma}} = \boldsymbol{W} \frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \widehat{\boldsymbol{\beta}}} = \boldsymbol{W} \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

Hence the vector $\vec{\gamma}$ and the gradient $\frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \vec{\gamma}}$, are both an element of the null space of V^T. Finally we could also find different basis for $\boldsymbol{W}$, as long as it spans the null space of V^T and is independent of the column vectors of V.

To construct an orthonormal basis for the null space of V^T, we first must state that we need $p(g-1)$ basis vectors. We can construct this basis as follows, for a small example where $p = 2$ and $g = 3$:

$$\boldsymbol{W} = \begin{bmatrix} \frac{g-1}{g} & 0 & 0 & 0 \\ 0 & 0 & \frac{g-1}{g} & 0 \\ -\frac{1}{g} & \frac{m-1}{m} & 0 & 0 \\ 0 & 0 & -\frac{1}{g} & \frac{m-1}{m} \\ -\frac{1}{g} & -\frac{1}{m} & 0 & 0 \\ 0 & 0 & -\frac{1}{g} & -\frac{1}{m} \end{bmatrix}$$

Where $m = g - 1$, we can generalize this construction for larger matrices. We now have an orthogonal basis, but still not an orthonormal one. We can show without loss of generality that each vector of this basis has the following norm:

$$\|\vec{v}_i\|_2 = \sqrt{\frac{(g-1)^2}{g^2} + \frac{(g-1)}{g^2}} = \sqrt{\frac{g^2 - 2g + 1 + g - 1}{g^2}} = \sqrt{\frac{g-1}{g}}$$

Its reciprocal is given by:

$$\frac{1}{\|\vec{v}_i\|_2} = \sqrt{\frac{g}{g-1}}$$

We can now easily construct an orthonormal set, for which we give one simple example:

$$\vec{q}_i = \frac{1}{\|\vec{v}_i\|_2}\vec{v}_i = \sqrt{\frac{g}{g-1}}\begin{bmatrix}\frac{g-1}{g}\\0\\-\frac{1}{g}\\0\\-\frac{1}{g}\\0\end{bmatrix} = \begin{bmatrix}\sqrt{\frac{g-1}{g}}\\0\\-\frac{1}{\sqrt{g(1-g)}}\\0\\-\frac{1}{\sqrt{g(1-g)}}\\0\end{bmatrix}$$

Hence, we now have an orthonormal basis. By using this basis we can solve the invariance problem of the optimization, resulting in fast optimizations without convergence problems (recall that the log-likelihood is concave and now free from invariance problems).

Hessian:

An additional methodology used in regression is the hypothesis testing of the regression coefficients.

H_0: The regression coefficient $\hat{\beta}_{ij}$ is equal to zero
H_1: The regression coefficient $\hat{\beta}_{ij}$ is not equal to zero

We assume that the null-hypothesis is normally distributed and that we can approximate it with Student's t-distribution. To calculate the p-value we first must define the t-statistic and the degree of freedom:

$$t = \frac{\hat{\beta}_{ij}}{S.E.(\hat{\beta}_{ij})}, d.f. = n-2$$

From the Gaussian-Markov theorem for linear models we can show the following:

$$\widehat{\boldsymbol{\beta}} \sim N(\boldsymbol{\beta}, \sigma^2(X^TX)^{-1})$$

Let us define the Hessian:

$$\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = -(\boldsymbol{X}^*)^T \boldsymbol{Z} \boldsymbol{X}^*$$

Taking the negative of this matrix we gain the information matrix:

$$\boldsymbol{I}(\boldsymbol{\beta}) = -\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = (\boldsymbol{X}^*)^T \boldsymbol{Z} \boldsymbol{X}^*$$

An estimator for the covariance matrix is given by:

$$\boldsymbol{\Sigma(\beta)|_{\beta=\hat{\beta}} = I^{-1}(\beta)\big|_{\beta=\hat{\beta}} = \left((X^*)^T Z X^*\right)^{-1}\Big|_{\beta=\hat{\beta}}}$$

Hence the estimator of the covariance matrix is given by the inverse of the Hessian. We can now define the variance and standard error for each regression coefficient.

$$var(\hat{\beta}_i) = \left[\left((\boldsymbol{X}^*)^{\boldsymbol{T}} \boldsymbol{Z} \boldsymbol{X}^*\right)^{-1}\right]_{ii}$$

Its standard error is than given by:

$$S.E.(\hat{\beta}_i) = \sqrt{[((\boldsymbol{X}^*)^T \boldsymbol{Z} \boldsymbol{X}^*)^{-1}]_{ii}}$$

Where $(\boldsymbol{X}^*)^T \boldsymbol{Z} \boldsymbol{X}^*$ is the negative Hessian when we differentiate the log-likelihood function for $\boldsymbol{\beta}$. A problem arises when we invert this Hessian due to the fact that its rank is $p(g-1)$ and its dimensions are $pg \times pg$. Hence, some columns are linearly dependent, or equivalently some eigenvalues are zero resulting in a singular matrix. We solve this by taking the Moore-Penrose inverse of the Hessian to calculate the standard errors.

This problem does not arise when we optimize the log-likelihood function for the reparameterization, or equivalently optimizing with the parameter vector $\vec{\gamma}$. We can this in the following way:

$$\frac{\partial^2 l(\widehat{\boldsymbol{\beta}})}{\partial \vec{\gamma} \partial \vec{y}^T} = \frac{\partial}{\partial \vec{y}^T}\left(\frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \vec{y}^T}\right)^T = \left(\frac{\partial}{\partial \widehat{\boldsymbol{\beta}}}\frac{\partial \widehat{\boldsymbol{\beta}}}{\partial \vec{\gamma}}\right)^T \left(\frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \widehat{\boldsymbol{\beta}}^T}\frac{\partial \widehat{\boldsymbol{\beta}}^T}{\partial \vec{\gamma}}\right)^T = \boldsymbol{W}^T \frac{\partial^2 l(\widehat{\boldsymbol{\beta}})}{\partial \widehat{\boldsymbol{\beta}} \partial \widehat{\boldsymbol{\beta}}^T} \boldsymbol{W} = \boldsymbol{W}^T \frac{\partial^2 l(\boldsymbol{\beta})}{\boldsymbol{\partial \beta \partial \beta^T}} \boldsymbol{W}$$

Recall that $\frac{\partial^2 l(\boldsymbol{\beta})}{\boldsymbol{\partial \beta \partial \beta^T}}$ is a $pg \times pg$ matrix, when we multiply it with the $pg \times p(g-1)$ matrix $\boldsymbol{W}$ as we do we have the $p(g-1) \times p(g-1)$ matrix $\frac{\partial^2 l(\widehat{\boldsymbol{\beta}})}{\partial \vec{\gamma} \partial \vec{y}^T}$. An advantage of this symmetric Hessian matrix, is that it is invertible due to the fact that it has a rank of pg. This due to the undoing of the invariance property of the log-likelihood function. Additionally, the diagonal elements of the inverse of this information matrix are all strictly positive.

Proof:

$$\frac{\partial^2 l(\widehat{\boldsymbol{\beta}})}{\partial \vec{\gamma} \partial \vec{y}^T} = \boldsymbol{W}^T \frac{\partial^2 l(\boldsymbol{\beta})}{\boldsymbol{\partial \beta \partial \beta^T}} \boldsymbol{W} = -\boldsymbol{W}^T (\boldsymbol{X}^*)^T \boldsymbol{Z} \boldsymbol{X}^* \boldsymbol{W}$$

$$\boldsymbol{I(\beta) = W}^T (\boldsymbol{X}^*)^T \boldsymbol{Z} \boldsymbol{X}^* \boldsymbol{W}$$

We now show that the information matrix has strictly positive diagonal elements:

$$\boldsymbol{I(\beta) = W}^T (\boldsymbol{X}^*)^T \boldsymbol{Z} \boldsymbol{X}^* \boldsymbol{W} = \boldsymbol{W}^T (\boldsymbol{X}^*)^T \boldsymbol{Q} \boldsymbol{D}^{\frac{1}{2}} \boldsymbol{D}^{\frac{1}{2}} \boldsymbol{Q}^T \boldsymbol{X}^* \boldsymbol{W}$$

Making the substitution $\boldsymbol{A^T = W}^T (\boldsymbol{X}^*)^T \boldsymbol{Q} \boldsymbol{D}^{\frac{1}{2}}$, resulting in:

$$\boldsymbol{W}^T (\boldsymbol{X}^*)^T \boldsymbol{Q} \boldsymbol{D}^{\frac{1}{2}} \boldsymbol{D}^{\frac{1}{2}} \boldsymbol{Q}^T \boldsymbol{X}^* \boldsymbol{W} = \boldsymbol{A}^T \boldsymbol{A}$$

$$\boldsymbol{A^T A} = \begin{bmatrix} \vec{A}_1^{\,T} \\ \vec{A}_2^{\,T} \\ \vdots \\ \vec{A}_{p(g-1)}^{T} \end{bmatrix} \begin{bmatrix} \vec{A}_1 & \vec{A}_2 & \cdots & \vec{A}_{p(g-1)} \end{bmatrix}$$

From this matrix multiplication we can see that:

$$[\boldsymbol{A^T A}]_{ii} = \left\|\vec{A}_i\right\|_2^2$$

And

$$\left\|\vec{A}_i\right\|_2^2 = 0 \;\leftrightarrow\; \vec{A}_i = \vec{0}$$

Hence, the last equation is contradicting as there is no vector in $\boldsymbol{A}$, that is equal to the zero-vector. This implies that all diagonal elements are strictly positive.

L-BFGS-B:

Fundamental problems arise when the Hessian is quite large or ill-conditioned. In some applications in can occur that the number of features, p, and samples, n are quite large. This results in a matrix with large dimensions which takes long to construct, but even longer to calculate its inverse. In some cases the Hessian is even ill-conditioned, such that numerical errors lead to the declaration of singularity for this matrix while it should be invertible. To avoid these ill-conditioned and time-consuming problems we decided to make use of the Quasi-Newton algorithm Limited-memory BFGS algorithm (with possible bounds). L-BFGS uses the Broyden–Fletcher–Goldfarb–Shanno update to approximate the Hessian matrix. L-BFGS is particularly well suited for optimization problems with a large number of dimensions. This because L-BFGS never explicitly forms or stores the Hessian matrix, which can be quite expensive when the number of dimensions becomes large. Instead, L-BFGS maintains a history of the past m updates of the position $l(\boldsymbol{\beta})$ and gradient $\nabla l(\boldsymbol{\beta})$, where generally the history m can be short, often less than 10. These updates are used to implicitly do operations requiring the Hessian (or it's inverse). Hence we are formulating the log-likelihood and gradient of the log-likelihood in terms of the parameter vector $\vec{\varphi}$, to remove the invariance property of the log-likelihood function. We have reasons to believe that for large problems, which this algorithm is intentially constructed for, this algorithm is much faster but also uses less memory.

Results:

To test the algorithm we have used the following dataset:

Author:	L.Dyrskjot et al.
Type:	Three types of bladder cancer
Number of samples:	40
Number of genes/features:	3036
Number of classes:	3

To test the algorithm we have selected 100 genes to perform the multinomial logistic regression with. We also tested with conventional packages such as 'globaltest'. Regrettably, this algorithm did not

converge or the Hessian was not invertible most of the times. Additionally, our algorithm required less computing time than other algorithms when fitting a large model.

Prob Class 1	Prob Class 2	Prob Class 3	Label
6.62E-14	1.00E+00	3.28E-23	2
8.02E-14	1.00E+00	1.96E-17	2
4.26E-14	1.00E+00	1.19E-14	2
9.17E-22	1.00E+00	7.92E-18	2
1.47E-13	1.00E+00	3.72E-14	2
1.12E-31	1.00E+00	4.54E-25	2
7.23E-23	1.00E+00	6.58E-20	2
1.90E-14	1.00E+00	5.86E-15	2
6.74E-14	1.00E+00	1.27E-16	2
1.40E-13	1.00E+00	4.02E-15	2
2.10E-15	1.00E+00	1.85E-18	2
7.83E-15	1.00E+00	3.74E-14	2
1.52E-18	1.00E+00	1.94E-14	2
7.27E-22	1.00E+00	2.39E-18	2
8.64E-15	1.00E+00	5.35E-16	2
2.82E-39	1.00E+00	6.54E-29	2
4.05E-14	1.00E+00	6.58E-15	2
6.41E-16	1.00E+00	3.62E-15	2
1.31E-18	2.24E-14	1.00E+00	3
2.74E-24	5.85E-14	1.00E+00	3
1.00E+00	7.80E-14	8.60E-15	1
1.00E+00	5.64E-14	1.09E-17	1
1.00E+00	1.64E-13	7.62E-18	1
1.00E+00	1.07E-40	3.67E-31	1
1.00E+00	1.50E-13	1.88E-19	1
1.00E+00	8.00E-14	8.18E-18	1
1.36E-24	1.00E+00	9.21E-15	2
1.94E-18	1.00E+00	1.83E-15	2
1.32E-25	1.00E+00	9.62E-14	2
1.59E-14	1.00E+00	6.55E-14	2
1.06E-21	1.00E+00	1.38E-18	2
4.38E-14	1.00E+00	2.41E-14	2
4.23E-27	1.00E+00	8.88E-19	2
8.11E-14	1.00E+00	1.81E-14	2
1.87E-13	1.00E+00	2.11E-15	2
3.52E-28	1.00E+00	4.38E-22	2
5.47E-14	1.00E+00	4.42E-14	2
1.06E-20	1.00E+00	2.76E-14	2
2.76E-24	1.00E+00	1.00E-16	2
2.08E-13	1.00E+00	1.42E-15	2

Table 1: Probabilities inferred by the model

From Table 1 we can clearly see that the training samples are perfectly classified, which is due to the overfitting of the model. To resolve this issue we should integrate penalization into the fitting of the model. This way we avoid overfitting and perform a procedure somewhat similar to parameter selection. An additional advantage is that we can integrate group Lasso, a subject later discussed in this document.

Penalization

Introduction:

One of the fundamental problems when trying to fit the model with the ordinary multinomial logistic regression model, when then the number of features exceeds the number of observations, is that it could lead to an overfit. Hence, inclusion of a larger number of features greatly increases the complexity of the model which tends to generate a higher variance. The increase in complexity has the advantage that it decreases the systematic error (bias) of the classifier, while increasing its variance. Ultimately, it results in an increase in the prediction error on the test sample while decreasing the error for the training sample. Hence it would be incongruous to increase the complexity of the model, as seen in Figure 2.1. What we are trying to find is the maximum parsimony of the model such that the increase its performance and interpretability of the prediction rule. To find this maximum parsimony, multiple methodologies and criterions have been devised to generate the model to one's own liking.

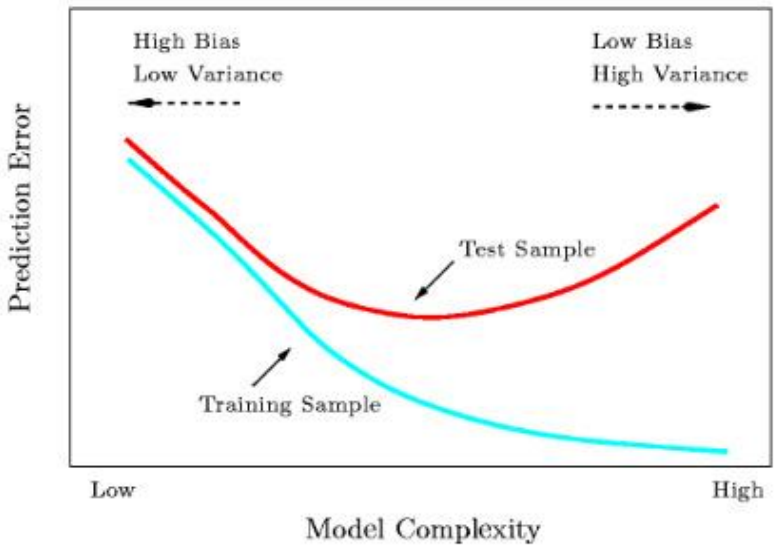


Figure 2.1: General behavior of test sample and training sample prediction error as the model complexity is varied.

The Bias-Variance decomposition

One framework we could use to understand the penalization methodologies is the Bias-Variance decomposition. In this section we shall explain this decomposition in an abstract way for the sake of clarity. Let us assume the following:

$$Y = f(\vec{x}) + \varepsilon$$

Where

$$E[\varepsilon] = 0$$
$$Var(\varepsilon) = \sigma_\varepsilon^2$$

We furthermore state the prediction or regression fit is given by:

$$\hat{Y} = \hat{f}(\vec{x})$$

For the sake for simplicity and generality we shall use the Loss function, also called squares-error loss:

$$L\left(Y, \hat{f}(\vec{x})\right) = (Y - \hat{f}(\vec{x}))^2$$

Than the expected prediction error, or Mean Squared Error, of the regression fit conditioned on the input point $X = \vec{x}_0$ is given by:

$$\begin{aligned}
&Err(\vec{x}_0) = E[(Y - \hat{f}(\vec{x}_0))^2 | X = \vec{x}_0] \\
&E[((\hat{f}(\vec{x}_0) - E[\hat{f}(\vec{x}_0)]) + (E[\hat{f}(\vec{x}_0)] - Y))^2 | X = \vec{x}_0] \\
&= E[(\hat{f}(\vec{x}_0) - E[\hat{f}(\vec{x}_0)])^2] + 2E[(\hat{f}(\vec{x}_0) - E[\hat{f}(\vec{x}_0)])(E[\hat{f}(\vec{x}_0)] - Y)] + E[(E[\hat{f}(\vec{x}_0)] - Y)^2] \\
&= E[(\hat{f}(\vec{x}_0) - E[\hat{f}(\vec{x}_0)])^2] + E[(E[\hat{f}(\vec{x}_0)] - f(\vec{x}_0) - \varepsilon)^2] \\
&= E[(\hat{f}(\vec{x}_0) - E[\hat{f}(\vec{x}_0)])^2] + E[E[\hat{f}(\vec{x}_0)]^2 - 2E[\hat{f}(\vec{x}_0)]f(\vec{x}_0) - 2E[\hat{f}(\vec{x}_0)]\varepsilon + f(\vec{x}_0)^2 + 2f(\vec{x}_0)\varepsilon \\
&\quad + \varepsilon^2] \\
&= E[(\hat{f}(\vec{x}_0) - E[\hat{f}(\vec{x}_0)])^2] + E[E[\hat{f}(\vec{x}_0)]^2 - 2E[\hat{f}(\vec{x}_0)]f(\vec{x}_0) + f(\vec{x}_0)^2 + \varepsilon^2] \\
&= E[(\hat{f}(\vec{x}_0) - E[\hat{f}(\vec{x}_0)])^2] + (E[\hat{f}(\vec{x}_0)] - f(\vec{x}_0))^2 + E[\varepsilon^2] \\
&= Var\left(\hat{f}(\vec{x}_0)\right) + Bias(\hat{f}(\vec{x}_0))^2 + \sigma_\varepsilon^2
\end{aligned}$$

We have shown that the prediction error consists out of an irreducible error (σ_ε^2), the variance of the predictor and the squared bias. Finally, we need to link the model complexity to the number of features or parameters. To show this we will make use of the theory of linear regression given as follows:

$$\begin{aligned}
\vec{Y} &= X\vec{\beta} + \vec{\varepsilon} \\
\hat{Y} &= X\hat{\beta} \\
\hat{\beta} &= \underset{\beta}{\operatorname{argmin}} \left\| \vec{Y} - X\vec{\beta} \right\|_2^2 \\
\hat{\beta} &= (X^T X)^{-1} X^T \vec{Y} \\
\hat{Y} &= X\hat{\beta} = X = (X^T X)^{-1} X^T \vec{Y} = H\vec{Y}
\end{aligned}$$

We have seen in the last chapter that this Hat matrix is idempotent and symmetric. We can now show the following:

$$Var(\hat{Y}) = Var(X\hat{\beta}) = Var(H\vec{Y}) = HVar(X\vec{\beta} + \varepsilon)H = \sigma_\varepsilon^2 H$$

If we then would calculate the average error over our test sample we would get:

$$\frac{1}{N}\sum_{i=1}^{N} Err(\vec{x}_i) = \sigma_\varepsilon^2 + \frac{1}{N}\sum_{i=1}^{N}(f(\vec{x}_i) - E[\hat{f}(\vec{x}_i)])^2 + \frac{\sigma_\varepsilon^2}{N}\sum_{i=1}^{N} H_{ii}$$

One fundamental theorem in Linear algebra gives us:

$$\sum_{i=1}^{N} H_{ii} = trace(H) = \sum_{i=1}^{N} \lambda_i$$

Due to the fact that the Hat matrix is idempotent we know that the eigenvalues are exactly 0 or 1. We also know that p (the number of parameters or features) eigenvalues are given the number 1. This implies that:

$$\sum_{i=1}^{N} \lambda_i = p$$

And eventually the average error becomes:

$$\frac{1}{N}\sum_{i=1}^{N} Err(\vec{x}_i) = \sigma_\varepsilon^2 + \frac{1}{N}\sum_{i=1}^{N}(f(\vec{x}_i) - E[\hat{f}(\vec{x}_i)])^2 + \frac{p}{N}\sigma_\varepsilon^2$$

Here the model complexity is directly related to the number of parameters. Increasing the number of parameters implies that we are increasing the so-called in-sample error. To reduce this variance substantially at the cost of some bias we need to reduce the effective dimensionality or the number of parameters.

Ridge regression

Introduction:

First we must clarify why we should not be satisfied by a least squares solution:

a) Prediction accuracy: The least squares estimates often have a low bias but large variance. The prediction accuracy sometimes can be improved by shrinking or setting some regression coefficients to zero. By doing so we sacrifice a little bit of bias to reduce the variance of the predicted values, and hence may improve the overall prediction accuracy.
b) Interpretation: With a large number of predictors, we often would like to determine a smaller subset that exhibits the strongest effects.

Both issues are occurring due to multi-collinearity. In regression when several predictors are highly correlated, the issue of multi-collinearity occurs. In a regression model we expect a high variance (R^2) explained. The higher the variance explained, the better the model. In a model where collinearity exists we expect that the model parameters and the variance are inflated. The high variance is not explained

by independent good predictors, but is due to a misspecified model that caries mutually dependent and thus redundant predictors. To cope for these issues we can make use of Ridge regression on a continuous level.

Ridge regression shrinks the regression coefficients by imposing a penalty on their size. It does this by minimizing the following expression:

$$RSS(\hat{\beta}^{ridge}) = \underset{\beta}{argmax}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=0}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{P}\beta_j^2\} = (\vec{y} - X\vec{\beta})^T(\vec{y} - X\vec{\beta}) + \lambda\vec{\beta}^T\vec{\beta}$$

Just like the least squares solution we can use Matrix calculus to minimize this expression. By taking the derivative of the residual sum of squares in terms of the regression coefficients and setting this equals to zero. We can do this by first noting that the function we are trying to minimize is concave. After taking the derivative we get the following:

$$\frac{\partial RSS(\hat{\beta}^{ridge})}{\partial\beta} = 2X^TX\vec{\beta} - 2X^T\vec{y} + 2\lambda\vec{\beta} = 0$$
$$X^TX\vec{\beta} + \lambda\vec{\beta} = X^T\vec{y}$$
$$\hat{\beta}^{ridge} = (X^TX + \lambda I)^{-1}X^T\vec{y}$$

It is possible that there are many highly correlated variables in a linear regression model, these parameters can become poorly determined and exhibit high variance. A large positive coefficient for one variable can be canceled by a large negative coefficient due to another correlated variable. This can be prevented by imposing penalties on the size of the coefficients. Due to the ridge solutions not being equivariant under scaling of the inputs, one should first standardize the inputs. It should be clear that you don't want to penalize the constant term. Now that the inputs are centered we need to know an estimate for the constant term β_0. This can be estimated by the following formula:

$$\beta_0 = \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

It is noted in the book of Friedman J., et al. that the solution adds a positive constant to the diagonal of X^TX before inversion. This should make the problem nonsingular, even if X^TX is not of full rank, and it was the main motivation for Ridge regression when it was first introduced by Hoerl and Kennard in statistics. The basic idea behind this could be that one wants to move the Gerschgorin disks such that the matrix becomes nonsingular.

Definition: Let X be a real or complex $n \times n$ matrix and let r_i denote the sum of the absolute values of the off-diagonal entries in the *i*th row of X. That is, $r_i = \sum_{j\neq i}^{n}|x_{ij}|$. The *i*th Gerschgorin disk is the circular disk D_i in the complex plane with center x_{ii} and radius r_i. That is,

$$D_i = \{z\ in\ \mathbb{C}: |z - x_{ii}| \leq r_i\}$$

Gerschgorin's Disk Theorem: Let X be an $n \times n$ real or complex matrix. Then every eigenvalues of X is contained with a Gerschgorin disk.

Proof:

Let λ be an eigenvalues X with corresponding eigenvector $\vec{v}$. Let v_i be the entry of $\vec{v}$ with the largest absolute values. Then $X\vec{v} = \lambda\vec{v}$, the *i*th row of which is:

$$\sum_{j=1}^{n} x_{ij}v_j = \lambda v_i$$

Rearranging we have:

$$(\lambda - x_{ii})v_i = \sum_{j\neq i}^{n} x_{ij}v_j \quad => \quad \lambda - x_{ii} = \frac{\sum_{j\neq i}^{n} x_{ij}v_j}{v_i}$$

Because $v_i \neq 0$, we obtain:

$$|\lambda - x_{ii}| = \left|\frac{\sum_{j\neq i}^{n} x_{ij}v_j}{v_i}\right| = \frac{|\sum_{j\neq i}^{n} x_{ij}v_j|}{|v_i|} \leq \frac{\sum_{j\neq i}^{n} |x_{ij}v_j|}{|v_i|} = \frac{\sum_{j\neq i}^{n} |x_{ij}||v_j|}{|v_i|} \leq \sum_{j\neq i}^{n} |x_{ij}| = r_i$$

Because $|x_j| \leq |x_i| \; for \; j \neq i$. This means that the eigenvalues λ is contained within the Gerschgorin disk centered at x_{ii} with radius r_i. In Figure 2.2 we can see some examples of Gerschgorin disks in the complex plane.

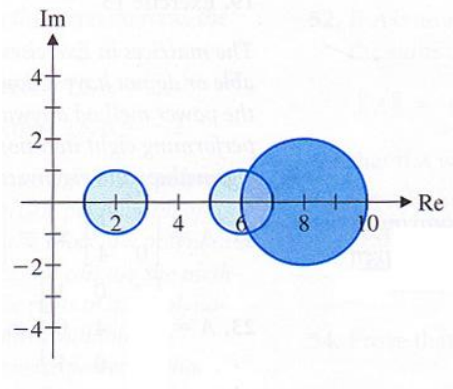


Figure 2.2: Gerschgorin disks in the complex plane

Because $X^T X$ is symmetric we also know that its eigenvalues are all real. This means that the Gerschgorin disks are just intervals on the real line. By adding constants to the diagonal entries of a matrix we are moving the Gerschgorin disks towards (larger) positive values while the radius stays the same. Eventually the matrix becomes what is called strictly diagonally dominant, where the absolute value of each diagonal entry is larger than the radius. This would mean that the matrix is always invertible as no disk overlaps the origin (0+i0). We know the following theorem:

$$\det(X) = \prod_{i=1}^{n} \lambda_i$$

By adding constants to the diagonal entries the eigenvalues will not become zero anymore and will turn the matrix in a nonsingular one.

As last part we can analyze the nature of Ridge regression. In the first part we have standardized the inputs which resulted in the new matrix X. Now we take the Singular Value Decomposition of this matrix, which is given by:

$$X = U\Sigma V^T$$
$$U: N \times p \; orthogonal \; matrix \; of \; which \; the \; columns \; spans \; the \; column \; space \; of \; X$$
$$\Sigma: P \times P \; diagonal \; matrix \; with \; the \; singular \; values$$
$$V: P \times P \; orthogonal \; matrix \; of \; which \; the \; column \; spans \; the \; row \; space \; of \; X$$

For the least squares solution this will give:

$$X\hat{\beta}^{LS} = X(X^TX)^{-1}X^T\vec{y} = U\Sigma V^T(VDV^T)^{-1}V\Sigma \mathrm{U}^{\mathrm{T}}\vec{y} = U\Sigma V^TVD^{-1}V^T\, V\Sigma \mathrm{U}^{\mathrm{T}}\vec{y} = \mathrm{U}\mathrm{U}^{\mathrm{T}}\vec{y}$$
$$= \sum_{\mathrm{i}=1}^{\mathrm{p}} \vec{\mathrm{u}}_{\mathrm{i}}\vec{\mathrm{u}}_{\mathrm{i}}^{\mathrm{T}}\vec{y}$$

This would indicate that the solution is a linear combination of the basis vectors spanning the column space. We can also do this for Ridge regression as it has an analytic solution:

$$X\hat{\beta}^{ridge} = X(X^TX + \lambda I)^{-1}X^T\vec{y} = U\Sigma V^T(VDV^T + \lambda I)^{-1}V\Sigma \mathrm{U}^{\mathrm{T}}\vec{y} = U\Sigma V^T(VDV^T + \lambda VV^T)^{-1}V\Sigma \mathrm{U}^{\mathrm{T}}\vec{y} =$$
$$U\Sigma V^T(V(D + \lambda I)V^T)^{-1}V\Sigma \mathrm{U}^{\mathrm{T}}\vec{y} = U\Sigma V^TV(D + \lambda I)^{-1}V^TV\Sigma \mathrm{U}^{\mathrm{T}}\vec{y} = U\Sigma(D + \lambda I)^{-1}\Sigma \mathrm{U}^{\mathrm{T}}\vec{y} =$$
$$\sum_{i=1}^{p} u_i \frac{\delta_i^2}{\delta_i^2 + \lambda} u_i^T\, \vec{y}$$

Here the values δ_i^2 are the singular values squared which are just the eigenvalues of X^TX. We can clearly see that the basis vectors of the column space get shrunken. It is also clear from the formula that the lower the value δ_i^2 the more shrunken the basis vector gets. The basic idea behind Ridge regression is that when one centers the input vector on can perform principal component analysis on the matrix. The principal components are in this case the basis vectors that span the column space of the centered matrix X. This method wants to preserve the column vectors which exhibit the most variance according to the eigenvalues. The directions with the smallest variance get shrunken the most by this method.
As last step we also have the effective degrees of freedom statistic for a given λ, denoted by df(λ). This can be calculated as follows:

$$df(\lambda) = tr(X(X^TX + \lambda I)^{-1}X^T) = \sum_{i=1}^{p} u_i \frac{\delta_i^2}{\delta_i^2 + \lambda} u_i^T = \sum_{i=1}^{p} \frac{\delta_i^2}{\delta_i^2 + \lambda}$$

This is equal to the amount of retained variance of the basis vectors of the column space. If we would set $\lambda = 0$ we would just get the least squares dimension, namely p.

We can also place the ridge regression methodology in the Bias-Variance framework by noting that:

$$Var(\hat{y}) = Var\left(X\hat{\beta}\right) = Var(X(X^TX + \lambda I)^{-1}X^T\vec{y}) = Var(H\vec{y}) = Hvar(\vec{y})H = \sigma_\varepsilon^2 H^2$$

The first thing we should now note about the projection matrix H is that it is symmetric but no longer idempotent. To see the contribution of the variance to the in-sample error we need to decompose the projection matrix by the singular value decomposition.

$$H^2 = U\Sigma(D + \lambda I)^{-1}\Sigma U^T U\Sigma(D + \lambda I)^{-1}\Sigma U^T = U\Sigma(D + \lambda I)^{-1}\Sigma^2(D + \lambda I)^{-1}\Sigma U^T$$
$$= U\Sigma(D + \lambda I)^{-1}D(D + \lambda I)^{-1}\Sigma U^T$$

Then the diagonal entries of this squared projection matrix is given by:

$$[H^2]_{ii} = [U\Sigma(D + \lambda I)^{-1}D(D + \lambda I)^{-1}\Sigma U^T]_{ii} = \vec{u}_i \frac{\delta_i^4}{(\delta_i^2 + \lambda)^2}\vec{u}_i^t = \frac{\delta_i^4}{(\delta_i^2 + \lambda)^2}$$

We can clearly see the following equality:

$$\delta_i^4 \leq (\delta_i^2 + \lambda)^2$$
$$\delta_i^2 \leq \delta_i^2 + \lambda$$

We know that the, due to the reason that the regularization parameter λ is always non-negative, that the fraction is always smaller than one, except for the case that the regularization parameter is exactly zero. For the in-sample error this implies the following:

$$\frac{1}{N}\sum_{i=1}^{N} Err(\vec{x}_i) = \sigma_\varepsilon^2 + \frac{1}{N}\sum_{i=1}^{N}(f(\vec{x}_i) - E[\hat{f}(\vec{x}_i)])^2 + \frac{\sigma_\varepsilon^2}{N}\sum_{i=1}^{N} H_{ii}^2$$
$$\frac{1}{N}\sum_{i=1}^{N} Err(\vec{x}_i) = \sigma_\varepsilon^2 + \frac{1}{N}\sum_{i=1}^{N}(f(\vec{x}_i) - E[\hat{f}(\vec{x}_i)])^2 + \frac{\sigma_\varepsilon^2}{N}\sum_{i=1}^{p} \frac{\delta_i^4}{(\delta_i^2 + \lambda)^2}$$

We are thus performing a summation over all predictors with fractions all smaller or equal to zero thereby reducing the variance. This fit has an additional estimation bias, due to the fact that it is not the closes fit in the model space. On the other hand, it has smaller variance. If the decrease in variance exceeds the increase in the squared bias, it is worthwhile as shown in Figure 2.3.

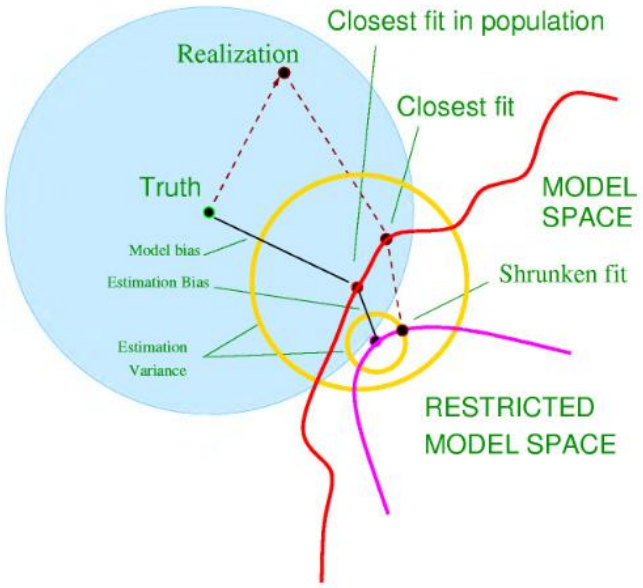


Figure 2.3: Behavior of the model variance and bias when applying penalization. The model space is the set of all possible configurations of the model. The closest fit is generated based upon the training data, but does not necessarily implies to best fit for the entire population (or samples from it). By penalization we can generate fits with possible better prediction probabilities.

Multinomial logistic regression:

It was worthwhile to explain ridge regression in a more abstract sense to truly understand the theory underlying this technique, but for this project we need to integrate it in the multinomial logistic regression framework. The dichotomous case has been developed by S. Le Cessie, et al. and has been promoted as good prediction model in a multitude of different fields of research, such as classification in microarray analysis. First we need to define the log-likelihood function we need to minimize:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \left\{ l(\vec{\beta}) + \frac{\lambda}{2} \sum_{i=1}^{p \cdot g} \beta_i^2 \right\} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \sum_{s=1}^{g} y_{is} (\eta_{is} - \ln(\sum_{t=1}^{g} e^{\eta_{it}})) + \frac{\lambda}{2} \sum_{i=1}^{p \cdot g} \beta_i^2 \right\}$$

To find the gradient of this log-likelihood function with respect to the beta coefficients we can show the procedure without loss of generality for one partial derivative.

$$\frac{\partial}{\partial \beta_{kh}} l(\boldsymbol{\beta}) = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \frac{\partial}{\partial \beta_{kh}} \sum_{s=1}^{g} y_{is} \left(\eta_{is} - \ln \left(\sum_{t=1}^{g} e^{\eta_{it}} \right) \right) + \frac{\lambda}{2} \sum_{i=1}^{p \cdot g} \frac{\partial}{\partial \beta_{kh}} \beta_i^2 = \sum_{i=1}^{n} \boldsymbol{X}_{ik} (y_{ih} - \mu_{ih}) + \lambda \beta_{kh}$$

Using the concatenated notation of the last chapter we can show that the gradient is given by:

$$\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = (\boldsymbol{X}^*)^T (\boldsymbol{y} - \boldsymbol{\mu}) + \boldsymbol{\lambda \beta}$$

We should note that the regularization parameter $\boldsymbol{\lambda}$ is given in bold due to the fact that it is a block matrix with a specialized structure. This comes from the fact that we wish not to penalize the constant parameter of each regression coefficient vector. The matrix has the following structure:

$$\begin{bmatrix} \boldsymbol{I}^* & 0 & \cdots & 0 \\ 0 & \boldsymbol{I}^* & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \boldsymbol{I}^* \end{bmatrix}$$

$$\boldsymbol{I}^* = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & \lambda & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda \end{bmatrix}$$

Due to the fact that the gradient consists out of two additive terms one can write the Hessian as follows:

$$\frac{\partial^2 l(\boldsymbol{\beta})}{\boldsymbol{\partial\beta\partial\beta^T}} = -(\boldsymbol{X}^*)^T \boldsymbol{Z} \boldsymbol{X}^* + \boldsymbol{\lambda}$$

We can now go one to prove the concavity of the problem, under the assumption that the model matrix $\boldsymbol{X}^*$ has been orthogonalized. This also implies that the cross product of a particular column with itself is equal to one and with another is equal to zero, hence resulting into a semi-orthogonal matrix.

$$-(\boldsymbol{X}^*)^T \boldsymbol{Z} \boldsymbol{X}^* + \lambda \boldsymbol{I} = (\boldsymbol{X}^*)^T(-\boldsymbol{Z} + \lambda \boldsymbol{I})\boldsymbol{X}^* = (\boldsymbol{X}^*)^T \boldsymbol{Q}(-\boldsymbol{D} + \lambda \boldsymbol{I})\boldsymbol{Q}^T \boldsymbol{X}^*$$
$$= (\boldsymbol{X}^*)^T \boldsymbol{Q}(-\boldsymbol{D} + \lambda \boldsymbol{I})^{\frac{1}{2}}(-\boldsymbol{D} + \lambda \boldsymbol{I})^{\frac{1}{2}} \boldsymbol{Q}^T \boldsymbol{X}^*$$

We now define $\boldsymbol{K} = (-\boldsymbol{D} + \lambda \boldsymbol{I})^{\frac{1}{2}} \boldsymbol{Q}^T \boldsymbol{X}^*$**:**

$$(\boldsymbol{X}^*)^T \boldsymbol{Q}(-\boldsymbol{D} + \lambda \boldsymbol{I})^{\frac{1}{2}}(-\boldsymbol{D} + \lambda \boldsymbol{I})^{\frac{1}{2}} \boldsymbol{Q}^T \boldsymbol{X}^* = \boldsymbol{K^T K}$$

Hence under orthogonality of the model matrix the problem is concave.

Identifiability:

In the ordinary multinomial logistic regression model we have seen that there are identifiability problems due to the fact that addition of particular vectors to the regression coefficient vectors leads to the same log-likelihood function value. This implies that the log-likelihood function has an optimum which at a particular direction has the same evaluation. This ultimately results that Newton-Raphson algorithm runs into non-convergent behavior, resulting into the break-down of the algorithm. To solve this problem we once again introduce the Omega transformation matrix $\boldsymbol{W}$**.** We first should identify the problem. Recall that the parameter vector is defined as a concatenated version of all regression coefficient vectors or conveniently called long notation:

$$\boldsymbol{\beta} = \begin{bmatrix} \vec{\beta}_1 \\ \vec{\beta}_2 \\ \vdots \\ \vec{\beta}_g \end{bmatrix}$$

Each regression coefficient vector then consists out of the unpenalized constant parameter and the p predictor parameters:

$$\vec{\beta}_i = \begin{bmatrix} \beta_{i0} \\ \beta_{i1} \\ \vdots \\ \beta_{ip} \end{bmatrix}$$

If we were to add the same values to each regression coefficient vector as seen in last chapter the log-likelihood function would remain the same, but the penalization function in the log-likelihood function changes and gives a different evaluation value.

$$\frac{\lambda}{2}\sum_{i=1}^{p\cdot g} \beta_i^2 \neq \frac{\lambda}{2}\sum_{i=1}^{p\cdot g} (\beta_i + c_i)^2$$

This would not be the case for the unpenalized constant coefficient parameter. If we would add the same value to each constant term we would get exactly the same log-likelihood function evaluation.

$$\vec{\beta}_i = \begin{bmatrix} \beta_{i0} \\ \beta_{i1} \\ \vdots \\ \beta_{ip} \end{bmatrix} + \begin{bmatrix} c \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

This eventually runs into identifiability problems. Unlike the multinomial logistic regression model case we now have $(p \cdot g) - 1$ unique parameters and the model is overfitted by one parameter. To resolve this issue we have constructed a new Omega matrix with the rank $(p \cdot g) - 1$. Where the gradient is ones more given by:

$$\frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \vec{\gamma}} = \frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \widehat{\boldsymbol{\beta}}} \frac{\partial \widehat{\boldsymbol{\beta}}}{\partial \vec{\gamma}} = \boldsymbol{W^T} \frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \widehat{\boldsymbol{\beta}}} = \boldsymbol{W^T} \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

And the Hessian by:

$$\frac{\partial l(\widehat{\boldsymbol{\beta}})}{\partial \vec{\gamma} \partial \vec{\gamma}^T} = \boldsymbol{W} \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta^T}} \boldsymbol{W}^T$$

Double cross-validation or Model selection:

Next to finding the optimal regression coefficient vector we also should find the most optimal regularization parameter λ. We should develop criterions to perform model selection and afterwards

perform model assessment to estimate the generalization error. Let us first define the training error rate:

$$\overline{err} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(\vec{x}_i))$$

This will be less than the true error:

$$Err = E\left[L\left(Y, \hat{f}(X)\right)\right]$$

Due to the fact that the same data is used to fit the model and asses the error. The model adapts to the training data, and hence the training error will be too optimistic of the generalization error Err. The phenomena of optimism is best understood in the in-sample error definition:

$$Err_{in} = E_y[\frac{1}{N}\sum_{i=1}^{N} E_{Y^{new}}[L(Y_i^{new}, \hat{f}(x_i))]]$$

The notation Y^{new} indicates that we are observing N new response values of the trainings points x_i. We define the optimism as the expected difference between Err_{in} and the training error $\overline{err}$.

$$op = Err_{in} - E_y[\overline{err}] = E_y[\frac{1}{N}\sum_{i=1}^{N} E_{Y^{new}}\left[L\left(Y_i^{new}, \hat{f}(x_i)\right)\right] - \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(\vec{x}_i))]$$

$$\frac{1}{N}\sum_{i=1}^{N} E_y[E_{Y^{new}}\left[L\left(Y_i^{new}, \hat{f}(x_i)\right)\right] - L(y_i, \hat{f}(x_i))]$$

It can be shown that for the squared loss function the optimism is given by:

$$op = \frac{2}{N}\sum_{i=1}^{N} Cov(y_i, \hat{y}_i)$$

This gives the following important formula:

$$Err_{in} = E_y[\overline{err}] + \frac{2}{N}\sum_{i=1}^{N} Cov(y_i, \hat{y}_i)$$

For the linear regression model it can even be shown that:

$$Cov(Y, \hat{Y}) = E[Y\hat{Y}^T] - E[Y]E[\hat{Y}]^T = (E[YY^T] - E[Y]E[Y]^T)H = Var(Y)H$$

$$\sum_{i=1}^{N} Cov(\hat{y}_i, y_i) = p\sigma_\varepsilon^2$$

Hence the in-sample error becomes:

$$Err_{in} = E_y[\overline{err}] + 2\frac{p}{N}\sigma_\varepsilon^2$$

Using this framework an obvious way to estimate the prediction error would be to estimate the optimism and add to it the training error. For this methodology we have developed methodologies such as the Akaike Information Criterion and Cross-Validation. The new form of the in-sample error is given by:

$$\widehat{Err_{in}} = \overline{err} + \widehat{op}$$

Where $\widehat{op}$ is an estimate of the optimism. The Akaike information criterion can generally be used if the a log-likelihood loss function is used. It relies on a relation that holds asymptotically when the number of observation $N \to \infty$.

$$-2E[\ln P_{\hat{\beta}}(Y)] \approx -\frac{2}{N}E[l(\hat{\beta})] + 2\frac{d}{N}$$

When the model complexity comes in to play we need to incorporate the regularization parameter λ. For this set of models we define:

$$AIC(\lambda) = \overline{err}(\lambda) + 2\frac{d(\lambda)}{N}\hat{\sigma}_\varepsilon^2$$

The function $AIC(\lambda)$ is an estimation of the test error curve, and we need to find the regularization parameter λ that minimized this function. An example of such a curve is portrayed in Figure 2.4.

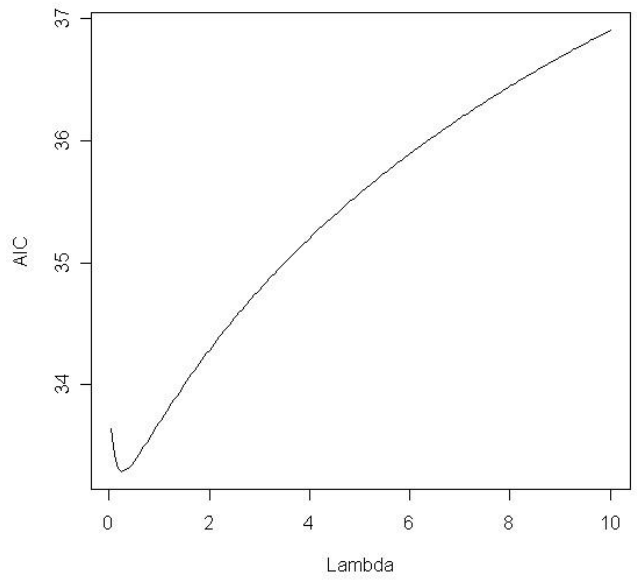


Figure 2.4: Akaike Information Curve being a function of the regularization parameter λ.

To calculate the AIC we must first define an appropriate Hat matrix for the penalized logistic regression. This is given by:

$$H = WX(X^T WX + \lambda I)^{-1} X^T$$

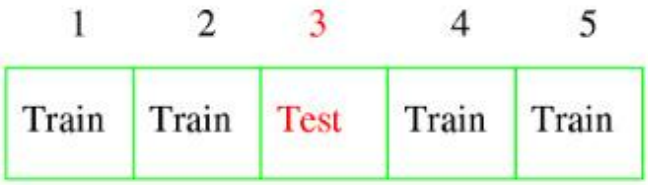
To estimate a good λ, we can use the following formula:

$$AIC(\lambda) = -2l(\vec{\beta}) + 2edf(\lambda) = 2(edf(\lambda) - l(\vec{\beta}))$$

Where $edf(\lambda) = tr(H)$. One of the main advantages is that the AIC can be calculated very fast when the optimal regression coefficient vector has been found.

Cross-Validation:

One of the most widely used methods is the cross-validated method. Ideally we would have a large enough dataset such that we can create a large training set and put away an unused test set that we can use later one for the estimation of the generalization error. Alas, this is not always possible and for such cases one could use cross-validation. One of the most used variants is called K-fold cross-validation. We split the data into K roughly equal sized parts. If we would for example take K=5, we could partition the dataset as follows:



For the third part, we would fit the model based upon K-1 parts of data as training set and calculate the prediction error when predicting the third part of the data. We do this for each part and combine the K estimates. Let us assume that we have an indexing function $g: \{1, \cdots, N\} \mapsto \{1, \cdots, K\}$ which assigns a membership to each observation in a randomized way. Denote by $\hat{f}^{-k}(x)$ the fitted function, trained without part k. Then the cross-validation estimation based upon the training data and the regularization parameter λ is given by:

$$CV(\lambda) = \frac{1}{K} \sum_{i=1}^{N} L(y_i, \hat{f}^{-k(i)}(x_i))$$

This cross-validation function estimates the test error curve and we should find the regularization parameter that minimized this function. It is also possible to define the number of parts equal to the number of observations; we also call this conveniently the leave-one-out cross-validation variant. The disadvantage of cross-validation is that it takes a lot of time to calculate it and it takes even more time to optimize the test error curve. The main advantage with respect to the Akaike Information Criterion is that is not bases on asymptotical assumption and is in cases where there is no time limitation a more congruent methodology.

The Brent Algorithm:

Although we can use different kinds of estimates for the test error curve we still need to find the minimum. One possibility is to let the use define a grid such he or she sees where approximately the optimum lies. In the case of Ridge regression this is a good method as it is a smooth function and has one global optimum (recall that the penalized log-likelihood function is concave). Due to this smoothness we could also devise methods to find the minimum automatically. For this we have implemented the BRENT algorithm. A downside of the developed model selection algorithms is that there mathematics cannot be used for differentiation; hence non differential algorithms must be used. The BRENT is a root finding algorithm that is based on root bracketing, bisection and inverse quadratic interpolation. With a slight modification that can be found in numerous books one can use it to find the optimum of a function. For this sole reason and its speed and accuracy we have implemented this algorithm additionally. Pre-liminary results have already shown that algorithm is precise and takes small amount of time to find the neighborhood of the minimum.

Results:

We expect that the implementation shrinks the regression coefficients to zero as the regularization parameter λ is increased. It is also well-known that correlated predictors are shrunk to zero simultaneously. One disadvantage is that the Ridge penalization never shrinks all regression coefficients exactly to zero, which only happened when we put the regularization parameter on infinity which lies outside of the practical scope. We furthermore expect that the framework makes convenient use of the bias-variance framework to increase the performance of the estimation of the prediction probabilities. Figure 2.5 shows an example what happened if we were to increase the regularization parameters. Please mind that the regression coefficients as shrunk according to the effective degrees of freedom which decreases as the regularization parameter increases. It clearly shows that the regression coefficients shrink together to zero.

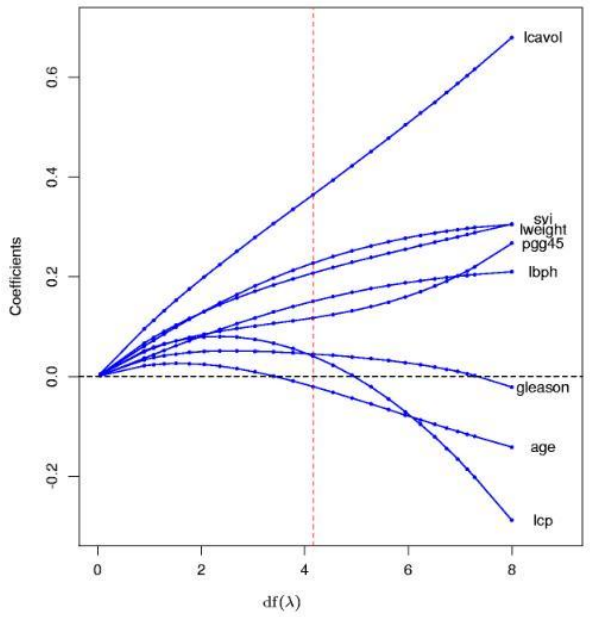


Figure 2.5: Shrinkage of the regression coefficients as the effective degrees of freedom is decreased.

To show some pre-liminary results, we yet again made use of the bladder cancer dataset:

Author:	L.Dyrskjot et al.
Type:	Three types of bladder cancer
Number of samples:	40
Number of genes/features selected:	200/3036
Number of classes:	3

Here we have set the regularization parameter λ to one to see what happens. We clearly see that the regression coefficients have shrunken to zero (not shown), but another clear distinction is that the estimated probabilities show less evidence of overfit. To estimate the test error we have used the LOOCV which was estimated to be approximately 0.25. We have also used the BRENT algorithm to find the optimal lambda which was estimated to be 34.2857433463778 with a LOOCV estimated test error of 0.199000474817524.

0.091077	0.907036	0.001887	2
0.045411	0.951008	0.003581	2
0.051887	0.945831	0.002282	2
0.01481	0.975183	0.010006	2
0.020055	0.96695	0.012995	2
0.001175	0.998105	0.00072	2
0.003067	0.995291	0.001642	2
0.010391	0.985692	0.003917	2
0.038143	0.957301	0.004556	2
0.026307	0.96915	0.004543	2
0.002629	0.996048	0.001323	2
0.014063	0.962696	0.023241	2
0.002857	0.994861	0.002282	2
0.007431	0.989935	0.002634	2
0.008014	0.978485	0.013501	2
0.000858	0.997976	0.001166	2
0.04285	0.944497	0.012653	2
0.003688	0.995531	0.00078	2
0.008515	0.051947	0.939538	3
9.67E-05	0.052804	0.947099	3
0.887736	0.105338	0.006926	1
0.943279	0.054486	0.002235	1
0.831453	0.161441	0.007106	1
0.994679	0.00441	0.000911	1
0.881429	0.116126	0.002445	1
0.918146	0.080522	0.001332	1
0.000648	0.983615	0.015737	2
0.003391	0.995069	0.001541	2
0.002145	0.958031	0.039824	2
0.011659	0.972815	0.015526	2
0.003229	0.985171	0.011601	2
0.003356	0.993904	0.00274	2
0.001366	0.99103	0.007604	2
0.008999	0.983022	0.00798	2
0.029504	0.966321	0.004175	2
0.009077	0.174932	0.815991	2
0.015123	0.965516	0.019361	2
0.000831	0.997599	0.00157	2

0.00117	0.988914	0.009916	2
0.059458	0.90541	0.035132	2

This table provides us the classification based upon the grade/malignancy of the tumor sample, but the authors did also provide us with other classification criteria based on tumor morphology or genetic aberrations. To infer the fit of the model we performed double cross-validation by using the BRENT algorithm, with LOOCV, to find the most optimal regularization parameter. In this case the optimal regularization parameter was found to be 2.8226047667724 with an estimated test error of 0.147079164231827. This is a general increase in prediction power when compared to the paper from which this dataset originates. The authors achieved a prediction accuracy of roughly 75%. We should that we do not have a secondary dataset to infer the generalization error and that we can only make predictions based upon an estimated test error. We should also note that we now also have some uncertainty about our classification, something that cannot be achieved by some other classification methods such as Support Vector Machine (SVM).

1	2	3	Type
0.915159	0.016466	0.068375	1
0.916726	0.016134	0.06714	1
0.930751	0.008705	0.060544	1
0.923985	0.030532	0.045484	1
0.967394	0.013726	0.01888	1
0.966659	0.012354	0.020987	1
0.962801	0.016905	0.020294	1
0.977591	0.0109	0.011509	1
0.94094	0.015673	0.043387	1
0.929132	0.022736	0.048132	1
0.962508	0.022277	0.015215	1
0.01994	0.962672	0.017388	2
0.041344	0.946907	0.011749	2
0.075804	0.880244	0.043952	2
0.004397	0.991352	0.004251	2
0.009161	0.988587	0.002252	2
0.002275	0.982955	0.014771	2
0.011658	0.9847	0.003642	2
0.022417	0.971368	0.006215	2
0.01516	0.976694	0.008145	2
0.033912	0.007665	0.958423	3
0.055509	0.008991	0.9355	3
0.034911	0.003004	0.962086	3
0.021306	0.005104	0.973591	3
0.098492	0.011948	0.889561	3

0.05631	0.01475	0.92894	3
0.001744	0.00723	0.991026	3
0.004391	0.001129	0.99448	3
0.004243	0.000453	0.995304	3
0.041223	0.001142	0.957635	3
0.007635	0.001647	0.990718	3
0.006906	0.020466	0.972628	3
0.002175	0.010441	0.987384	3
0.006037	0.000183	0.99378	3
0.000397	0.000202	0.999401	3
0.004974	0.001264	0.993762	3
0.001993	0.001629	0.996378	3
0.002194	0.027644	0.970161	3
0.019161	0.003074	0.977766	3

Lasso regression

Introduction:

As we have indicated before the data analyst should not be satisfied with the Ordinary Least Squares estimates upon which the multinomial logistic regression is based. This has two good reasons:

1. Prediction Accuracy: The ordinary least squares estimates often have low bias, but high variance as was shown in the bias-variance framework. The prediction accuracy can sometimes by shrinking or setting some regression coefficients exactly to zero. By doing so we might sacrifice a little bit of (squares) bias for a substantial decrease in the variance of the predicted probabilities. Hence, may improve the overall prediction accuracy.
2. Interpretation: With a large number of predictors, we often would like to determine a smaller subset that exhibits the strongest effect with respect to the outcome or classification. These are major issues when using the multinomial logistic regression model with or without the L_2 penalization. In these cases all the predictors will retain a regression coefficient larger than zero. Furthermore, correlated predictors will be shrunk together to zero instead of retaining one and discarding the others.

Other techniques such as subsets selection indeed give an interpretable prediction rule but can be highly variable as it is a discrete process. Small changes in the data can already result in different models being selected and can result in a substantial decrease in the prediction accuracy. To solve these issues Robert Tibshirani (1996) developed a technique called lasso, for 'least absolute shrinkage and selection operator'. In this case it shrinks some coefficients and sets other exactly to zero, and hence tries to retain the good features of both subset selection and ridge regression. First we should define the lasso minimization function:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}\{l(\vec{\beta})\} \; subject\ to \sum_j |\beta_j| \leq s$$

Where $l(\vec{\beta})$is the log-likelihood function evaluated at $\vec{\beta}$ and is subjected to an L_1 penalization. Another more appropriate definition is given by:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}\{l(\vec{\beta}) - \lambda\|\boldsymbol{\beta}\|_1\}$$

The two definitions are equivalent, as the latter definition can be constructed as the Lagrange multiplies version of the former optimization problem under the Katush-Kuhn-Tucker (KKT) conditions. It is a nice regularization method as it performs variable selection as well as shrinkage. The amount of shrinkage is dependent on the magnitude of regularization employed and is very useful for generating interpretable prediction rules for high dimensional data in which the number of features exceeds the number of observations. Figure 2.5 portrays the main difference between lasso and ridge penalization when using linear models.

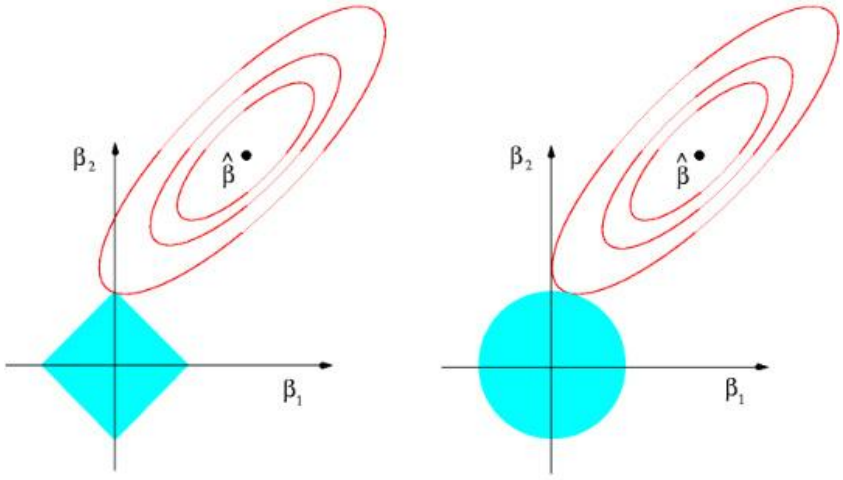


Figure 2.5: Estimation of the regression coefficients for the lasso (left) and ridge regression (right). Shown are the contour lines of the error and constraint functions. It can clearly be seen that the lasso shrinks the parameter β_1 to zero whereas ridge regression does not.

Many efficient algorithms have been developed to minimize both optimization functions. In the case of linear models quadractic programming and even exact solution algorithms have been developed, but for generalized linear models, such as multinomial logistic regression, it will be computationally more demanding. Various path-based algorithms have been developed generating accurate results. These algorithms are based on high-dimensional paths being a function of the regularization parameter λ. These paths are normally piecewise linear giving the advantage that the regression coefficients for a particular regularization parameter can be calculated on the fly, but for generalized linear models these paths exhibit a somewhat linear piecewise function, as shown in Figure 2.6. Hence, in this case most algorithms make use of linear approximation and the estimation of the parameters.

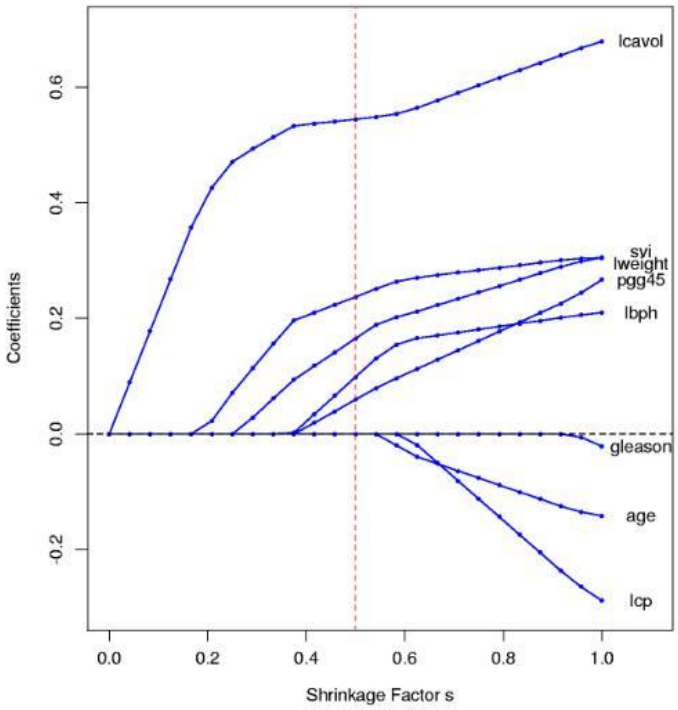


Figure 2.6: Solution paths of the lasso coefficients as the shrinkage factor is varied.

Problem description:

Let us define the target function we would like to minimize:

$$l_{pen}(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) - \lambda \sum_{i=1}^{p \cdot g} |\beta_i|$$

It consists out of two terms: The first being the log-likelihood of the concatenated regression coefficient vector $\boldsymbol{\beta}$ and the penalization function $P(\boldsymbol{\beta}) = -\lambda \sum_{i=1}^{p \cdot g} |\beta_i|$. The log-likelihood function is of the class C^2, implying that it is at least twice differentiable everywhere, and concave. Alas, the penalization function is less well-behaved: it is concave and continuous, but it is only differential at all points except for $\beta_i = 0$ for all i. Hence, the target function being the sum of two concave functions is itself concave although it is not strictly concave. This results in the possibility that the target function can have a flat top; being a contiguous optimum consisting out of multiple points. Furthermore it is not differentiable everywhere. This implies that we have $3^{p \cdot g} - 1$ continuous subspaces which are differentiable. A meaningful question would be how we could modify the target function such that we can make use of Quasi-newton methods for which we need not to calculate the Hessian exactly. One method that we have implemented is projecting every continuous subspace into one subspace where the target function is continuous. To do this we first must define the positive part and negative part function:

$$\beta_i^+ = \max(\beta_i, 0)$$
$$\beta_i^- = -\min(\beta_i, 0)$$

We should note that the positive as well as the negative part are both non-negative. They both are always equal or larger than zero. Using these functions we can define:

$$\beta_i = \beta_i^+ - \beta_i^-$$
$$|\beta_i| = \beta_i^+ + \beta_i^-$$

Hence, we can reparameterize the model such that every regression coefficient can be expresses by the summation of two positive parameters. By creating twice as much parameters we can map the loose continuous subspaces into higher dimension subspace where every parameter is non-negative. For the target function we could write the reparameterization as follows:

$$l_{pen}(\boldsymbol{\beta}^+, \boldsymbol{\beta}^-) = l(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-) - \lambda \sum_{i=1}^{p \cdot g} \beta_i^+ + \beta_i^- = l(\boldsymbol{\beta}) - \lambda \sum_{i=1}^{2 \cdot p \cdot g} \beta_i^*$$

The additional advantage is that we can calculate the gradient because we now have mapped to a higher dimensional subspace with a continuous function.

$$\frac{\partial l(\beta)}{\partial \beta^+} = \frac{\partial l(\beta)}{\partial \beta} \frac{\partial \beta}{\partial \beta^+} = \frac{\partial l(\beta)}{\partial \beta}$$
$$\frac{\partial l(\beta)}{\partial \beta^-} = \frac{\partial l(\beta)}{\partial \beta} \frac{\partial \beta}{\partial \beta^-} = -\frac{\partial l(\beta)}{\partial \beta}$$
$$\frac{\partial P(\beta)}{\partial \beta} = -\lambda$$

The entire gradient of all the new parameters is given by:

$$\frac{\partial l(\beta)}{\partial \beta^*} = \begin{bmatrix} \frac{\partial l(\beta)}{\partial \beta} \\ -\frac{\partial l(\beta)}{\partial \beta} \end{bmatrix} - \lambda \boldsymbol{I}^*$$

We should note that we have stated the new parameters of the reparameterization should all be non-negative. To account for this issue we specifically have employed the L-BFGS-B algorithm as it can put a lower or upper bound on the parameters to be optimized. For this reason we have put a lower bound of zero on each parameter.

Cross-validation:

To accommodate the data analyst with an automatic algorithm for the search of the optimal regularization parameter we have once more employed the Cross-validation methodology. The BRENT algorithm is once more used to find the optimum. One down-side is that the estimated test error curve is not always as smooth as we would like it to be. Furthermore it could be possible that multiple local optima exist. An example of this non-smooth behavior is portrayed in Figure 2.7.

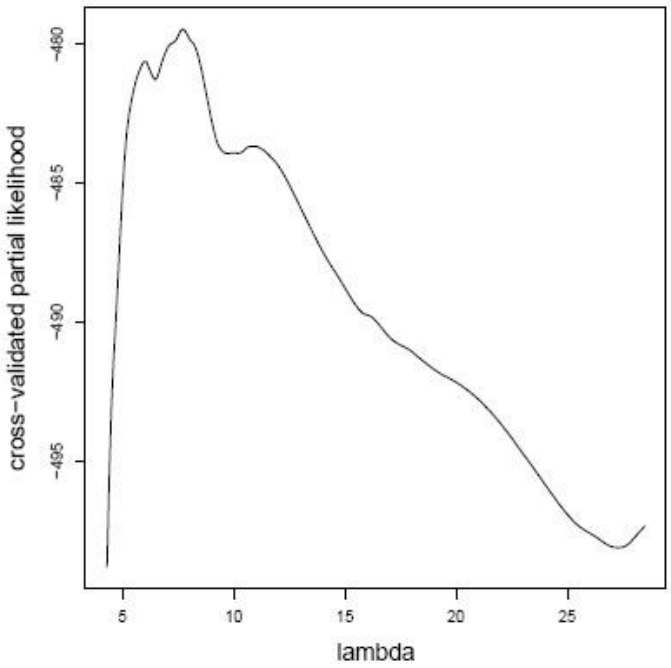


Figure 2.7: The cross-validated partial log-likelihood as a function of the regularization parameter. It clearly shows the non-smooth behavior of the estimated test error curve.

One advantage that we can use in our optimization is that when:

$$\lambda \geq [l(\boldsymbol{\beta})]_i \; for \; all \; i$$

We can say that the regression coefficient vector becomes:

$$\boldsymbol{\beta} \equiv \mathbf{0}$$

Hence we can calculate the maximum of the regularization parameter for which all predictors are discarded from the model. To accommodate the user with the possibility to search for the optimum we advise them to first use a grid search to find a section on the line of the regularization parameter for which the neighborhood of the optimum is located.

Results:

We will test the multinomial logistic regression model penalized by a L_1 penalty ones again on the Bladder cancer dataset. The first thing we need to perform is to find a segment on the regularization parameter line for which the CV curve is somewhat optimal. First we have defined a grid between 0.1 and 1.1 with a step size of 0.1. The results are shown in the right image of Figure 2.8, it is clear that we need to extend our window to find the optimum. We then generated a grid between 0.1 and 10.1 with a step size of 0.5. The result of this grid is portrayed in the right image of Figure 2.8. It clearly shows that the minimum is found somewhat around $\lambda = 2$. For this reason we are applying the BRENT algorithm between $\lambda = 1$ and $\lambda = 3$.

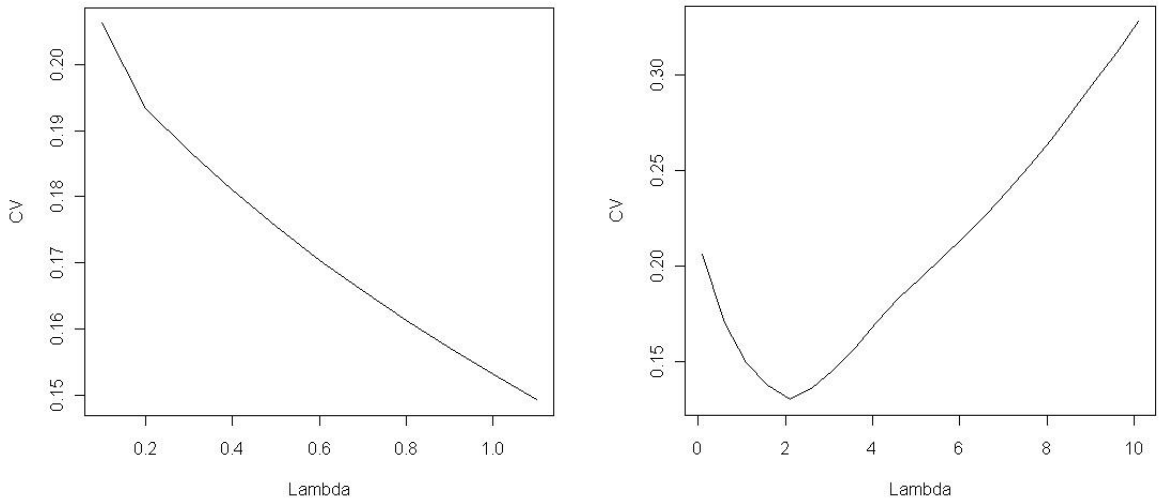


Figure 2.8: (Right) The Cross-validation curve between 0.1 and 1.1. (Left) The Cross-validation curve between 0.1 and 10.1.

After only ten evaluations the algorithm converges at a cross-validation value of 0.1299629. It immediately becomes clear that using the full model leads to an overfit, due to the fact that at this regularization level only 3 to 4 predictors are retained of the initial 405 predictors. We can see from the table of estimated prediction probabilities that some observations would be misclassified. That is not a problem, as we are trying to fit the model such that it would perform best on the population or at least a new sample of the observations.

1	2	3	Type
0.373356	0.220487	0.406157	1
0.752456	0.062806	0.184738	1
0.655487	0.038351	0.306162	1
0.710162	0.069747	0.22009	1
0.689049	0.157069	0.153882	1
0.823122	0.036727	0.140151	1
0.821358	0.078928	0.099714	1
0.807941	0.15022	0.041839	1
0.781624	0.08937	0.129006	1
0.541128	0.116254	0.342619	1
0.792307	0.103733	0.10396	1
0.189501	0.757636	0.052862	2
0.337993	0.555242	0.106765	2
0.288671	0.601237	0.110092	2
0.003649	0.985853	0.010498	2
0.077597	0.89671	0.025693	2

0.048145	0.896675	0.05518	2
0.04502	0.947885	0.007095	2
0.188947	0.770344	0.04071	2
0.088549	0.703892	0.207559	2
0.221005	0.016985	0.76201	3
0.138496	0.014215	0.847289	3
0.227286	0.013778	0.758936	3
0.136596	0.027618	0.835786	3
0.438225	0.194671	0.367103	3
0.350974	0.036839	0.612187	3
0.008515	0.039923	0.951562	3
0.037552	0.012405	0.950043	3
0.010974	0.00701	0.982016	3
0.099581	0.01848	0.881939	3
0.036124	0.008699	0.955177	3
0.058265	0.108023	0.833712	3
0.013029	0.021878	0.965093	3
0.019446	0.002765	0.977789	3
0.014683	0.003296	0.982021	3
0.013289	0.061389	0.925322	3
0.079471	0.056182	0.864347	3
0.0281	0.109958	0.861942	3
0.048964	0.005191	0.945844	3

Elastic net

Introduction:

The advantage of penalized the regression coefficients with a L_1 norm is that it improves the interpretability over multinomial logistic regression and ridge regression. The downside is that is sometimes has a slight worse prediction accuracy than ridge regression. To cope with this problem H. Zou and T. Hastie (2003) developed and new penalization function called the elastic net. This is an algorithm we additionally added to our package. As a continuous shrinkage method, ridge regression achieves its better prediction performance through a bias-variance trade-off. However, ridge regression cannot produce a parsimonious model, for it always keeps all the predictors in the model. The lasso technique should solve these issues. Although the lasso has shown success in many situations, it has some of its limitations:

1. In the $p > n$ case, the lasso selects at most n variables before it saturates, which seems to be a limiting feature for a variable selection method.
2. If there is a group of variables with high pairwise correlations, then the lasso tends to select only one variable from the group and does not case which one is selected.

3. In the case that $n > p$, if there are high correlations between predictors, it has been empirically observed that the prediction performance of the lasso is dominated by ridge regression.

The first two situations make the lasso as variable selection method in some cases incongruous, especially when we want to interpret the prediction rule. In the classification of biological samples we could stumble upon pathways, of which the genes are frequently pairwise correlated. These genes could be classified as groups. In good variable selection methods the trivial genes would be discarded whereas a group should be preserved once one gene among them is selected. For the grouped variables situation, the lasso is not a good algorithm. To improve these penalization phenomena we have implemented the elastic net algorithm. Which performs automatic variables election and continuous shrinkage, and it can select groups of correlated variables automatically.

Elastic net:

First we need to introduce the target function we need to minimize:

$$\widehat{\boldsymbol{\beta}} = \underset{\beta}{\operatorname{argmin}}\{l_{net}(\boldsymbol{\beta})\} = \underset{\beta}{\operatorname{argmin}}\left\{l(\boldsymbol{\beta}) + \frac{\lambda_2}{2}\|\boldsymbol{\beta}\|_2^2 + \lambda_1\|\boldsymbol{\beta}\|_1\right\}$$

We see that log-likelihood function now is penalized by a L_2 and a L_1 norm. We could see this target function as a somewhat intermediate between the lasso and ridge regression, although that is not entirely the case as the model is vulnerable to the scaling of the predictors. Figure 2.9 illustrates the contour plots of the penalization curves. The round contour plot is generated by ridge regression whereas the contour plot with strict corners is generated by the lasso. The last contour plot is generated by the elastic net penalization function and we can clearly see that it is a somewhat intermediate between the two other contour plots.

Due to the reason that the lasso penalization function is added to the target function we once more make use of the reparameterization.

$$|\beta_i| = \beta_i^+ + \beta_i^-$$
$$\beta_i^2 = (\beta_i^+ - \beta_i^-)^2 = (\beta_i^+)^2 - 2\beta_i^+\beta_i^- + (\beta_i^-)^2$$
$$\widehat{\boldsymbol{\beta}}_{elas} = \underset{\beta}{\operatorname{argmin}}\left\{l(\boldsymbol{\beta}) + \frac{\lambda_2}{2}\sum_{i=1}^{p\cdot g}(\beta_i^+ - \beta_i^-)^2 + \lambda_1\sum_{i=1}^{2\cdot p\cdot g}\beta_i^*\right\}$$

We can now also define the gradient:

$$\frac{\partial l(\beta)}{\partial \beta^+} = \frac{\partial l(\beta)}{\partial \beta}\frac{\partial \beta}{\partial \beta^+} = \frac{\partial l(\beta)}{\partial \beta}$$
$$\frac{\partial l(\beta)}{\partial \beta^-} = \frac{\partial l(\beta)}{\partial \beta}\frac{\partial \beta}{\partial \beta^-} = -\frac{\partial l(\beta)}{\partial \beta}$$
$$\frac{\partial P_2(\beta)}{\partial \beta^+} = \frac{\partial P_2(\beta)}{\partial \beta}\frac{\partial \beta}{\partial \beta^+} = \frac{\partial P_2(\beta)}{\partial \beta} = -\lambda_2\boldsymbol{\beta}$$
$$\frac{\partial P_2(\beta)}{\partial \beta^-} = \frac{\partial P_2(\beta)}{\partial \beta}\frac{\partial \beta}{\partial \beta^-} = -\frac{\partial P_2(\beta)}{\partial \beta} = \lambda_2\boldsymbol{\beta}$$

$$\frac{\partial P_1(\beta)}{\partial \beta^*} = -\lambda_1$$

So that the gradient becomes:

$$\frac{\partial l(\beta)}{\partial \beta^*} = \begin{bmatrix} \frac{\partial l(\beta)}{\partial \beta} \\ -\frac{\partial l(\beta)}{\partial \beta} \end{bmatrix} - \lambda_2 \begin{bmatrix} -\boldsymbol{\beta} \\ \boldsymbol{\beta} \end{bmatrix} - \lambda_1 \boldsymbol{I}^*$$

We should now note that the target function is dependent on two different regularization parameters, namely λ_1 and λ_2. Hence, the cross-validation function is also dependent on these two regularization parameters. To solve the issue for finding the optimal regularization parameters we first employ a grid search such that the data analyst has the general idea where the optimal values are somewhat located. For the bladder cancer dataset we have created a grid as illustrated in Figure 2.10(A) and 2.10(B).

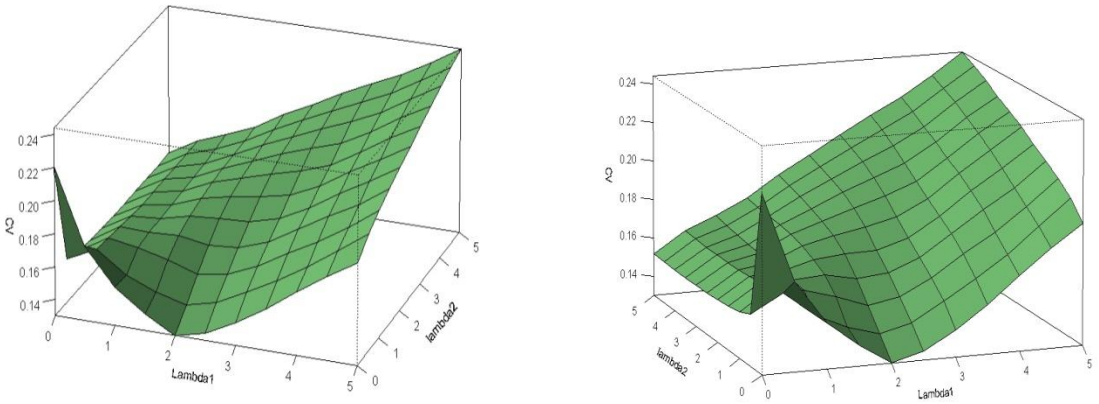


Figure 2.10: (A) The cross-validation surface dependent on the two regularization parameters (B) The same cross-validation surface, but now from another angle.

It is clear from both images that the surface is not smooth in the direction of the λ_1 parameter, but tends to be smooth in the direction of the λ_2 parameter. To solve this issue we once again have implemented the BRENT algorithm, but now that algorithm is applied in a two-state procedure. First the direction of λ_1 is optimized by setting λ_2 to zero and afterwards the function is optimized in the direction of λ_2. Although we cannot guarantee that the global optimum is found, we expect that this methodology generally gives a good estimate of the optimal regularization parameters in an automatic way. If one wants to find the exact estimate, the only option would be to apply the grid search in a finer resolution.

Results:

We applied the elastic net to the bladder cancer dataset with the same set of genes used for all the other algorithms. We first applied the grid search on this dataset and the results are illustrated in Figure 2.10. One can clearly see that the optimum is somewhat around $\lambda_1 \approx 2.0$ and $\lambda_2 \approx 0.0$. It seems that

for this particular dataset it is best to not use the ridge regression penalization in terms of the estimated test error. Hence, in this case we will get the same results as with the lasso.

1	2	3	Type
0.373356	0.220487	0.406157	1
0.752456	0.062806	0.184738	1
0.655487	0.038351	0.306162	1
0.710162	0.069747	0.22009	1
0.689049	0.157069	0.153882	1
0.823122	0.036727	0.140151	1
0.821358	0.078928	0.099714	1
0.807941	0.15022	0.041839	1
0.781624	0.08937	0.129006	1
0.541128	0.116254	0.342619	1
0.792307	0.103733	0.10396	1
0.189501	0.757636	0.052862	2
0.337993	0.555242	0.106765	2
0.288671	0.601237	0.110092	2
0.003649	0.985853	0.010498	2
0.077597	0.89671	0.025693	2
0.048145	0.896675	0.05518	2
0.04502	0.947885	0.007095	2
0.188947	0.770344	0.04071	2
0.088549	0.703892	0.207559	2
0.221005	0.016985	0.76201	3
0.138496	0.014215	0.847289	3
0.227286	0.013778	0.758936	3
0.136596	0.027618	0.835786	3
0.438225	0.194671	0.367103	3
0.350974	0.036839	0.612187	3
0.008515	0.039923	0.951562	3
0.037552	0.012405	0.950043	3
0.010974	0.00701	0.982016	3
0.099581	0.01848	0.881939	3
0.036124	0.008699	0.955177	3
0.058265	0.108023	0.833712	3
0.013029	0.021878	0.965093	3
0.019446	0.002765	0.977789	3
0.014683	0.003296	0.982021	3

0.013289	0.061389	0.925322	3
0.079471	0.056182	0.864347	3
0.0281	0.109958	0.861942	3
0.048964	0.005191	0.945844	3

Due to the reasons that this is not so spectacular we also used the other classification problem where we are separating the tumors based on malignancy. This cross-validation surface is shown in Figure 2.11, and it clearly shows that in the case the lasso regularization parameter λ_1 is best to set to zero. In this case we get the best results by using only ridge regression and hence we get the same results as in the previous chapters. In this dataset we clearly see that a combination of the two penalization functions is not preferred. We need to search for a dataset were this is not the case.

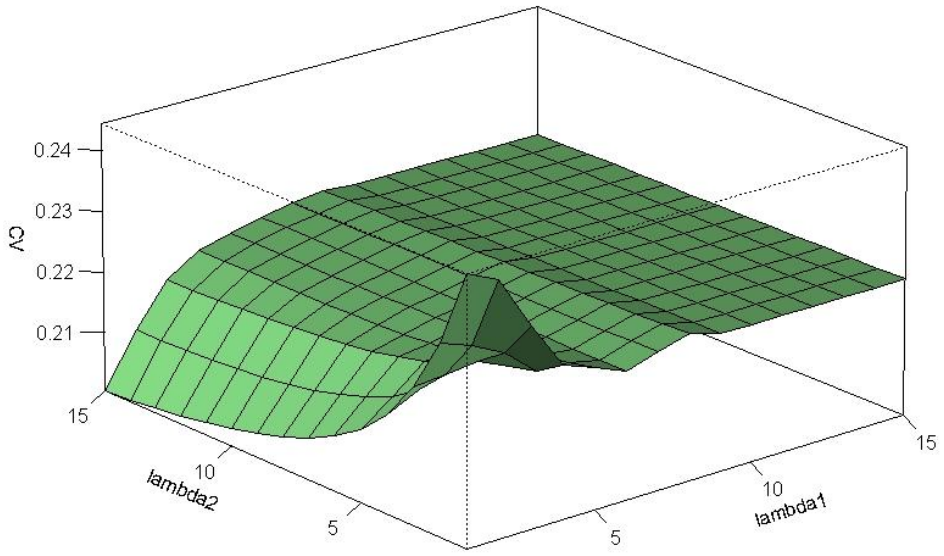


Figure 2.11: Cross-validation surface bases on the classification of the tumor grade of malignancy.

Group lasso

Introduction:

A problem occurring with the regular lasso algorithm is that it tends to retain or discard predictors, irrespective of all the other regression vectors for the other classes. In the Table below we see an excerpt of the prediction rules generated for a 4-class classification problem. As is obvious the Lasso penalization performs a selection on the individual covariates when dealing with a multi-class problem. It is quite unnatural to select regression coefficients instead of entire predictors. In the case of Lasso a predictor is automatically selected if one of its regression coefficients is selected. Inherently this leads to prediction rules with many zeros, but also the selection of more predictors than desired and maybe necessary.

	Other	T(15;17)	T(8;21)	INV(16)	Gene Symbol
1553588_at	9.55E-05	0	0	-0.0003	ND3
200026_at	9.91E-05	0	0	0	RPL34
200665_s_at	0	0	0	0.000659	SPARC
201324_at	-0.00024	0	0	0	EMP1
201360_at	-0.00014	0	0	0.000246	CST3
201432_at	0.00173	0	0	-0.00039	CAT
201502_s_at	0.000318	0	0	0	NFKBIA
201721_s_at	0	0	-0.00053	0	LAPTM5
202746_at	0	0	0	0.000388	ITM2A
202859_x_at	0	0	0.000122	0	IL8
202902_s_at	0	0	0	0.000201	CTSS
202917_s_at	0	0	0	0.00021	S100A8
203535_at	0	0	0	0.000762	S100A9

Table 1. Regression coefficients of a 4-class classification problem: Lasso has a tendency to set many regression coefficients to zero.

We wish to retain or discard an entire predictor. To accomplish this we are going to make use of the Group Lasso penalization in multinomial logistic regression. This algorithm is an extension of the regular Lasso and was developed by Yuan and Lin (2006) and Meier and Buhlmann (2008). The advantage of this penalization is that we can define grouping structures on which we can perform variable selection instead of single predictors. This could be for instance prior knowledge of pathways, or similarity in enzymatic function of the genes (based on Gene Ontology or KEGG annotation). In our case we wish to group the regression coefficients of one specific gene and perform the Group Lasso. In this case a particular gene is retained or discarded in all classes as their regression coefficient is put non-zero or zero simultaneously for all classes. The Group Lasso can be seen as an intermediate between Ridge and Lasso penalization. As illustrated in Figure 2.12, a group of regression coefficients is shrunk simultaneously to zero, when the regularization parameter is increased, similar to Ridge penalization. It also shows that as when the regression coefficients of a group are set to zero, it will remain zero.

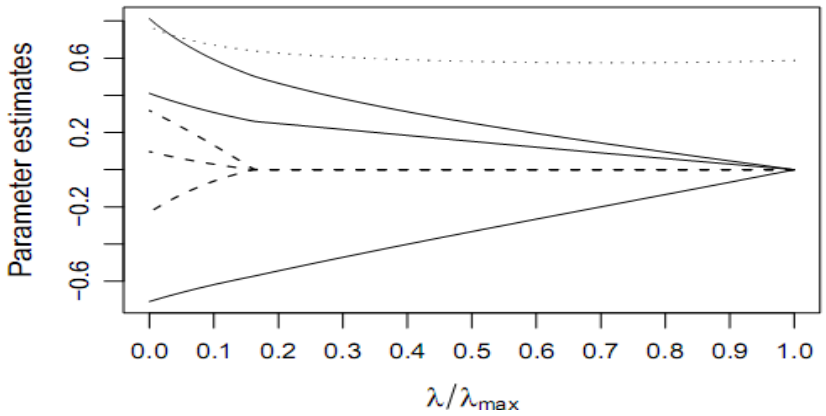


Figure 2.12: Behavior of regression coefficients in a group (Meier L. *et al*)

Group Lasso:

We should define the beta matrix, of which the columns consist of regression coefficient vectors for each class:

$$\tilde{\beta} = \begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1g} \\ \beta_{21} & \beta_{22} & & \beta_{2g} \\ \vdots & & \ddots & \vdots \\ \beta_{p1} & \beta_{p2} & \cdots & \beta_{pg} \end{bmatrix} = [\boldsymbol{\beta}_1 \quad \boldsymbol{\beta}_2 \quad \cdots \quad \boldsymbol{\beta}_g]$$

This beta matrix gives the opportunity to define many group structures, and was the underlying mechanism for the development of the Group Lasso. In this study, we would like to retain or discard each predictor; i.e. each row of this matrix, by setting all regression coefficients simultaneously unequal or equal to zero. This is accomplished by defining each row vector of regression coefficients as a group. Let us assume that we have a p-dimensional feature vector $\boldsymbol{x}_i \in \mathbb{R}^{\boldsymbol{p}}$, which consists out of J groups. Let us denote by df_j the degrees of freedom of group j, rewrite $\boldsymbol{x}_i = \left(\boldsymbol{x}'_{i,1}, \boldsymbol{x}'_{i,2}, \cdots, \boldsymbol{x}'_{i,J}\right)'$ and denote the group of variable $\boldsymbol{x}'_{i,j} \in \mathbb{R}^{\boldsymbol{df_j}}$, $j = 1, \cdots, J$. The regression coefficient vector is parameterized as $\boldsymbol{\beta}_t = \left(\beta_{0t}, \boldsymbol{\beta}_{1,t}, \boldsymbol{\beta}_{2,t}, \cdots, \boldsymbol{\beta}_{J,t}\right)'$, $t = 1, \cdots, G$. Given these groups we rewrite the logistic curve function as:

$$P(Y_i = s) = \mu_{is} = \frac{e^{\beta_{0s} + \sum_{j=1}^{J} \boldsymbol{x}'_{i,j} \boldsymbol{\beta}_{j,s}}}{\sum_{t=1}^{\boldsymbol{g}} e^{\beta_{0t} + \sum_{j=1}^{J} \boldsymbol{x}'_{i,j} \boldsymbol{\beta}_{j,t}}}$$

It is clear that under this new definition the logistic curve function has changed somewhat. Next point is to proof that this definition leads to the same logistic curve function under our defined group structure. Let us first denote the linear predictor $\eta_{is} = \beta_{0s} + \sum_{j=1}^{J} \boldsymbol{x}'_{i,j}\boldsymbol{\beta}_{j,s}$. To obtain the linear predictors for all observation over all classes we denote:

$$\begin{bmatrix} \eta_{11} \\ \vdots \\ \eta_{n1} \\ \vdots \\ \eta_{1g} \\ \vdots \\ \eta_{ng} \end{bmatrix} = \begin{bmatrix} \beta_{01} \\ \vdots \\ \beta_{01} \\ \vdots \\ \beta_{0g} \\ \vdots \\ \beta_{0g} \end{bmatrix} + \sum_{i=1}^{p} \begin{bmatrix} \boldsymbol{x_i} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{x_i} & & \vdots \\ & & \ddots & \\ \mathbf{0} & \cdots & & \boldsymbol{x_i} \end{bmatrix} \begin{bmatrix} \beta_{i1} \\ \beta_{i2} \\ \vdots \\ \beta_{ig} \end{bmatrix} = \begin{bmatrix} \beta_{01} + x_{11}\beta_{11} + \cdots + x_{1p}\beta_{p1} \\ \vdots \\ \beta_{01} + x_{n1}\beta_{11} + \cdots + x_{np}\beta_{p1} \\ \vdots \\ \beta_{0g} + x_{11}\beta_{1g} + \cdots + x_{1p}\beta_{pg} \\ \vdots \\ \beta_{01} + x_{n1}\beta_{1g} + \cdots + x_{np}\beta_{pg} \end{bmatrix}$$

Where $\boldsymbol{x}_i \in \mathbb{R}^{\boldsymbol{p}}$ is the *i*-th column of the design matrix. Hereby we show that the linear predictor for the group lasso under the proposed group structure is similar to the linear predictor for the multinomial logistic regression model. Hence, we proof:

$$\mu_{is} = \frac{e^{\beta_{0s} + \sum_{j=1}^{J} \boldsymbol{x}'_{i,j}\boldsymbol{\beta}_{j,s}}}{\sum_{t=1}^{\boldsymbol{g}} e^{\beta_{0t} + \sum_{j=1}^{J} \boldsymbol{x}'_{i,j}\boldsymbol{\beta}_{j,t}}} = \frac{e^{\boldsymbol{\beta_{0s}} + \boldsymbol{x}'_i\boldsymbol{\beta}_s}}{\sum_{t=1}^{\boldsymbol{g}} e^{\boldsymbol{\beta_{0t}} + \boldsymbol{x}'_i\boldsymbol{\beta}_t}}$$

The Group Lasso estimator β_λ is given by the maximizer of the function:

$$\ell_{glasso}(\boldsymbol{\beta}^*; \lambda) = \ell(\boldsymbol{\beta}^*) - \lambda \sum_{j=1}^{J} \left\| \boldsymbol{\beta}_j \right\|_2 = \ell(\boldsymbol{\beta}^*) - \lambda \sum_{j=1}^{p} \sqrt{\beta_{i1}^2 + \cdots + \beta_{ig}^2} = \ell(\boldsymbol{\beta}^*) - \psi(\boldsymbol{\beta}^*)$$

Hence, the penalty function sums the norm of each row vector of the beta matrix $\tilde{\beta}$. Note that Meier L. (2008) as well as Yuan M. (2006) integrate the square root of the degrees of freedom of each group in the summation. Given the current group structure, each group has the same degrees of freedom, thus the additional term is omitted. To optimize the penalized log-likelihood function, the low-memory BFGS algorithm Liu (1989) is used. The gradient of the penalized log-likelihood function is given by:

$$\frac{\partial \ell_{glasso}(\boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}^*} = \frac{\partial \ell(\boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}^*} - \lambda \frac{\partial \psi(\boldsymbol{\beta}^*)}{\partial \boldsymbol{\beta}^*},$$

where the gradient of the penalty function is defined as:

$$\frac{\partial \psi(\boldsymbol{\beta}^*)}{\partial \beta_{ij}} = \frac{\partial}{\partial \beta_{ij}} \left(\sqrt{\beta_{11}^2 + \cdots + \beta_{1g}^2} + \cdots + \sqrt{\beta_{p1}^2 + \cdots + \beta_{pg}^2} \right) = \frac{\beta_{ij}}{\sqrt{\beta_{i1}^2 + \cdots + \beta_{ig}^2}}$$

Optimizing the penalized log-likelihood function leads to major problems, as the function is only strictly convex and continuous in all subspaces of the regression coefficients. The derivative of the penalized log-likelihood function remains undefined when one of the regression coefficients equals zero. This is

issue is resolved by reparameterizing the model to a higher dimension where the function is strictly convex and continuous. The following reparameterization is proposed:

$$\beta_{ij} = \beta_{ij}^{+} - \beta_{ij}^{-}$$
$$\beta_{ij}^{+} = \max(\beta_{ij}, 0)$$
$$\beta_{ij}^{-} = -\min(\beta_{ij}, 0)$$
$$\beta_{ij}^{+} \geq 0$$
$$\beta_{ij}^{-} \geq 0$$

The reparameterization is realized by decomposing the individual regression coefficients into a positive part function (PPF) and a negative part function (NPF). These functions are constrained by the fact that each must be non-negative. For this reason we make use of the box constraints that can be set for the L-BFGS-B algorithm. Note that at the convergence either the PPF, NPF or both should be equal to zero. This reparameterization results in a model with twice as many parameters, which are restricted to a subspace of non-negative regression coefficients. As stated, in this single subspace the penalized log-likelihood function is strictly convex, continuous, and is differentiable in each internal point. Hence, instead of dealing with distinct continuous subspaces where the function is non-differentiable at their borders, i.e. when one of the regression coefficients is set to zero, we now have one subspace where the function is differentiable in its internal space. The log-likelihood gradient remains unchanged under the reparameterization, but the penalty function gradients are given by:

$$\frac{\partial \psi(\boldsymbol{\beta}^*)}{\partial \beta_{ij}^{+}} = \frac{\partial \psi(\boldsymbol{\beta}^*)}{\partial \beta_{ij}} \frac{\partial \beta_{ij}}{\partial \beta_{ij}^{+}} = \frac{\partial \psi(\boldsymbol{\beta}^*)}{\partial \beta_{ij}}$$
$$\frac{\partial \psi(\boldsymbol{\beta}^*)}{\partial \beta_{ij}^{-}} = \frac{\partial \psi(\boldsymbol{\beta}^*)}{\partial \beta_{ij}} \frac{\partial \beta_{ij}}{\partial \beta_{ij}^{-}} = -\frac{\partial \psi(\boldsymbol{\beta}^*)}{\partial \beta_{ij}}$$

A problem occurs when all the regression coefficients of a group become zero, as the penalty function is no longer differentiable. To solve this problem the following limit is taken for the sake of continuity:

$$\lim_{\beta_{ij} \to 0} \frac{\beta_{ij}}{\sqrt{\beta_{i1}^2 + \cdots + \beta_{ig}^2}} = 1, \quad if\ \beta_{i1} = \cdots = \beta_{i(j-1)} = \beta_{i(j+1)} = \cdots = \beta_{ig} = 0$$

Next to the reparameterization, the optimization of the penalized log-likelihood is also affected by a parameter identifiability problem. The penalty function $\psi(\boldsymbol{\beta}^*)$ consists of the norms of the row vectors of the beta matrix $\tilde{\beta}$. These norms are characterized by the squared regression coefficients β_{ij}^2, belonging to their respective groups. Under the reparameterization this squared regression coefficient is given by:

$$\beta_{ij}^2 = \left(\beta_{ij}^{+} - \beta_{ij}^{-}\right)^2 = {\beta_{ij}^{+}}^2 - 2\beta_{ij}^{+}\beta_{ij}^{-} + {\beta_{ij}^{-}}^2 ,$$

In this equation, multiple instances of β_{ij}^{+} or β_{ij}^{-} could give the exact same β_{ij}^2. This problem can be resolved by imposing a constraint on this equation. At convergence either the PPF, NPF or both should be equal to zero. This implies that the middle term of the factorization of β_{ij}^2 should be forced to be zero. This leads to the redefinition of equation above:

$$\beta_{ij}^2 = \beta_{ij}^{+2} + \beta_{ij}^{-2}$$

As we are trying to redefine the penalty function it is more appropriate to rewrite the penalized log-likelihood function:

$$\ell_{glasso}(\boldsymbol{\beta}^*; \lambda) = \ell(\boldsymbol{\beta}^*) - \lambda \sum_{j=1}^{J} \sqrt{\|\boldsymbol{\beta}_j^+\|_2^2 + \|\boldsymbol{\beta}_j^-\|_2^2}$$

It is easily shown through the triangle-inequality that:

$$\sqrt{\|\boldsymbol{\beta}_j^+\|_2^2 + \|\boldsymbol{\beta}_j^-\|_2^2} \geq \sqrt{\|\boldsymbol{\beta}_j^+ - \boldsymbol{\beta}_j^-\|_2^2} = \sqrt{\|\boldsymbol{\beta}_j^+\|_2^2 - 2(\boldsymbol{\beta}_j^+)^T \boldsymbol{\beta}_j^- + \|\boldsymbol{\beta}_j^-\|_2^2}$$

Hence, the redefinition of the penalty function $\psi(\boldsymbol{\beta}^*)$ is always larger or equal than its original definition. Given the inequality and the fact that either the PPF, NPF or both are zero at convergence, the redefined penalty function becomes equal to the original definition. By this redefinition we have solved the parameter identifiability problem and proven to be exactly the same as the original definition at convergence, we obtain the exact same prediction rules without convergence problems.

The table below illustrates an excerpt of the results from the same 4-class classification problem based on the modified Group Lasso. In comparison with Table 1 it immediately becomes clear that: (i) the number of predictors is decreased (ii) no regression coefficient of the retained predictors is set to zero, and (iii) the new group structure facilitates comparison of the regression coefficients between classes.

	Other	T(15;17)	T(8;21)	INV(16)	Gene Symbol
1553588_at	0.00018085	-8.59E-05	5.97E-05	-0.0001546	ND3
200665_s_at	-0.00014592	-1.73E-05	-7.23E-05	0.000235584	SPARC
201324_at	-0.00017254	1.18E-05	2.64E-05	0.000134331	EMP1
201360_at	-0.00020149	9.67E-06	-6.17E-05	0.000253532	CST3
201432_at	0.000946723	-0.00025838	-3.35E-05	-0.0006548	CAT
201502_s_at	0.000131047	-0.00012284	6.43E-05	-7.25E-05	NFKBIA
201721_s_at	0.000325746	1.74E-05	-0.00034902	5.93E-06	LAPTM5
202746_at	-0.00012466	1.67E-05	-0.00011551	0.000223436	ITM2A
202902_s_at	-7.06E-06	-1.00E-05	-5.58E-06	2.27E-05	CTSS
202917_s_at	-6.06E-05	-0.0001884	-8.16E-05	0.000330612	S100A8
203535_at	-0.00018007	-6.25E-05	-9.28E-05	0.000335433	S100A9

Table 2. Regression coefficients of a 4-class classification problem with the modified Group Lasso: The Group Lasso procedure produces sparser prediction rules. Furthermore it facilitates the comparison of regression coefficients between classes.

Results:

For most of the test results we would like to refer to the paper “Sparse multi-class prediction based on the Group Lasso in multinomial logistic regression” and its supplementary which can be found in the Chapter Paper on page 28. In this section we will discuss some results that are obtained, but not given in the paper.

Global test:

We have applied the global test for multinomial logistic regression (Goeman J.J., 2004) to investigate whether the fit of the model can discriminate the classes based on the given predictors, i.e. genes. This test can determine whether the global expression pattern of all genes is significantly related to the outcomes, i.e. class labels. For classification case 1, which consist out of classes with a favorable risk except for the ‘Other’ class it immediately clear that the multinomial logistic regression model can accurately discriminate the given classes as illustrated in Table 3.

Cohort 1				
p-value	Statistic	Expected	Std.dev	#Cov
6.91E-17	2.58	0.386	0.0975	54675

Table 3 Global test results: The results gives positive evidence that the classes can be accurately discriminated

We have also applied the global test on the second classification case where we are trying to discriminate classes based on the NPM1 and FLT3ITD mutation. We have applied the global test procedure on both cohorts to give positive evidence that the quality is similar. This is illustrated in Table 4. It is obvious on the bases of the p-values that the gene expression data contains information to discriminate the given classes, although less than in classification case 1.

Cohort 1				
p-value	Statistic	Expected	Std.dev	#Cov
3.64E-08	1.43	0.392	0.0869	54675
Cohort 2				
p-value	Statistic	Expected	Std.dev	#Cov
8.05E-08	1.36	0.385	0.0863	54675

Table 4 Global test results: The results gives positive evidence that the classes can be accurately separated and that the two cohorts are of similar quality.

Survival analysis:

In classification case 1 it became apparent that the CEBPA$^{double\text{-}mut}$ samples have a distinct and discriminative gene expression profile compared to the CEBPA$^{single\text{-}mut}$ and. CEBPA^{wt} samples. In this case all CEBPA$^{single\text{-}mut}$ are misclassified due to a weak gene expression pattern. To show that these groups are significantly different we have create Kaplan-Meier curves and used the pooled log-rank test to see if their respective distributions are significantly different. We used the statistical software package SPSS to accomplish this. Figure 2.13 illustrates the Kaplan-Meier curves and it can be seen clearly that the CEBPA$^{double\text{-}mut}$ samples have a higher survival probability after 5 years. The overall survival, Figure 2.13A, clearly shows a difference between the single- and the double-mutated samples. This is difference is significant as seen in Figure 2.14 (p=0.011). For the event-free survival (EFS) the double- and single-mutated samples also differ significantly as shown in Figures 2.13B and 2.15.

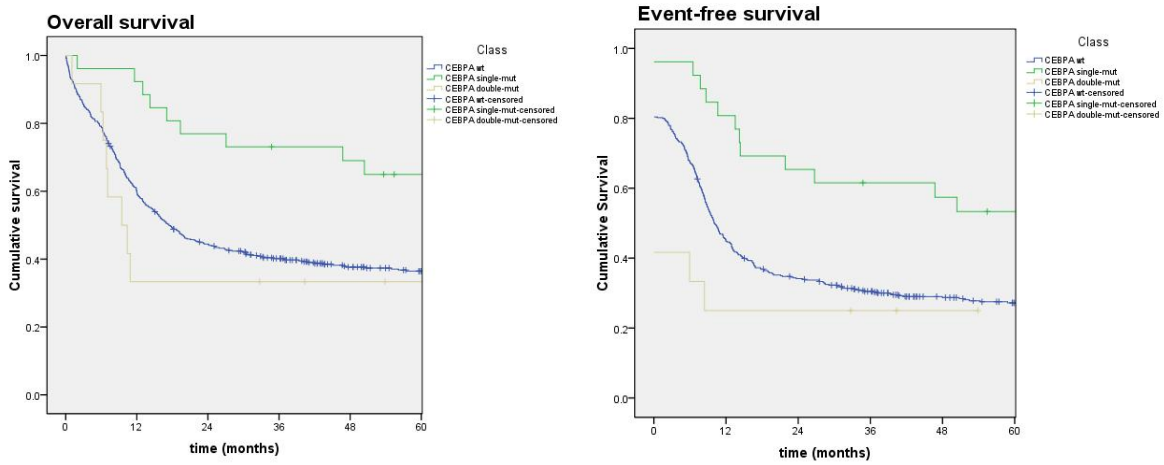


Figure 2.13. Overall survival and event-free survival.

C. Overall survival among CEBPA$^{double\text{-}mut}$ vs. CEBPA$^{single\text{-}mut}$ vs. CEBPA^{wt}, Log rank test, pooled: p=0.011
D. Event-free survival among CEBPA$^{double\text{-}mut}$ vs. CEBPA$^{single\text{-}mut}$ vs. CEBPA^{wt}, Log rank test, pooled: p=0.008

Overall Comparisons

	Chi-Square	df	Sig.
Log Rank (Mantel-Cox)	8.942	2	.011

Test of equality of survival distributions for the different levels of Class.

Figure 2.14. Pooled log rank test for the OS.

Overall Comparisons

	Chi-Square	df	Sig.
Log Rank (Mantel-Cox)	9.780	2	.008

Test of equality of survival distributions for the different levels of class.

Figure 2.15. Pooled log rank test for the EFS.

MAPK pathway:

For classification case 2 the retained predictors show an affinity for ribosomal, heatshock, immunoglobulin and HOX proteins. The HOX proteins have a heavy impact on the classification of the classes harboring samples with the NPM1 mutations. Many genes in the expression signature are related to processes of cellular stress, inflammation response and DNA repair mechanisms. For the large number of ribosomal genes we could give the following explanations: (i) it could be the case that DNA repair or cell homeostasis mechanisms are activated in the response to the abnormalities arising from cancer formation (ii) NPM1 is a known chaperon protein for ribosomal proteins in the nucleolus. The mutation dislocates NPM1 and could indicate that it is also a complement for the construction of the ribosomal proteins, although this is highly speculative. The inflammatory and immunoglobulin response could be induced through well-known MAPK-pathway. A substantial number of retained genes could be mapped back to the MAPK pathway (Figure 2.16, Ingenuity), which can initiate inflammatory and (anti)-apoptotic mechanisms. If these mutations truly initiate these mechanism is only speculative and should be distinguished on the basis of further and more elaborate research.

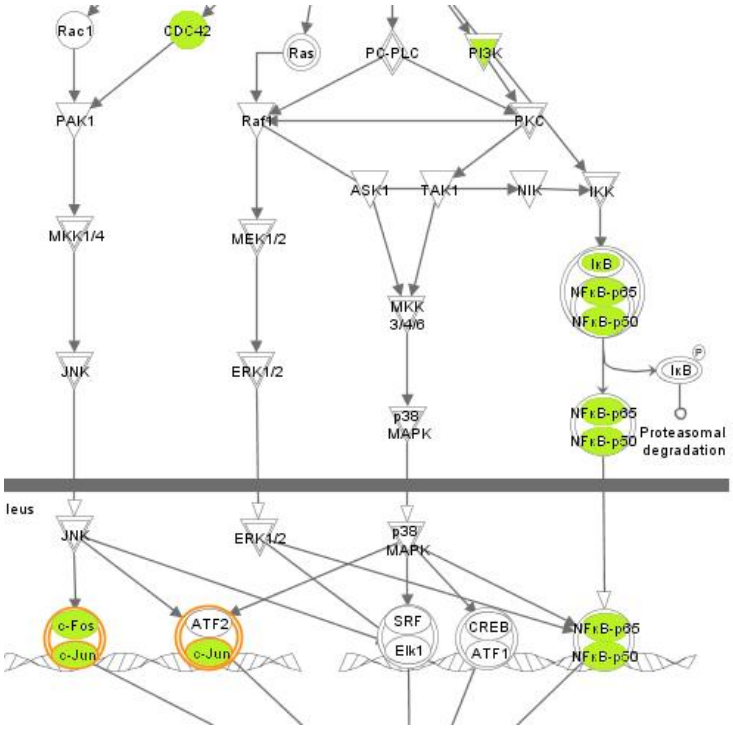


Figure 2.16 MAPK pathway: The constituents of the MAPK pathways are depicted in this figure. The green color indicates that the gene is retained in the gene signature.

References

le Cessie S., van Houwelingen JC. (1992), Ridge Estimators in Logistic Regression, *Applied statistics*, Volume 41(1), 191-201

Goeman JJ., le Cessie S. (2006), A goodness-of-fit test for multinomial logistic regression, *Biometrics,* Volume 62(4), 980-985

Krishnapuram B., Carin L., Figueiredo MAT., Hartemink AJ. (2005), Sparse Multinomial Logistic Regression: Fast Algorithms and Generalization Bounds, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Volume 27(6), pp. 957-968

Liu DC., Nocedal J. (1989), On the limited memory BFGS method for large scale optimization, *Mathematical programming 45*, 503-528

Meier L., van de Geer S. and Bühlman P (2008), The Group Lasso for logistic regression. *Journal of the Royal Statistical Society B 70(1)*, 53- 71

Park MY., Trevor H. (2007), L_1 Regularization Path Algorithm for Generalized Linear Models, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, Volume 69(4) , pp. 659-677(19)

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

Valk PJM., Verhaak RGW., Beijen MA., Erpelinck CAJ., Barjesteh van Waalwijk van Doorn-Khosrovani S., Boer JM., Beverloo HB., Moorhouse MJ., van der Spek PJ., Löwenberg B., Delwel R. (2004), Prognostically Usefule Gene-Expression Profiles in Acute Myeloid Leukemia*, New England Journal of Medicine 350*, 1617-1628

Wouters BJ., Löwenberg B., Erpelinck-Verschueren CAJ., van Putten WLJ., Valk PJM., and Delwel R. (2009), Double CEBPA mutations, but not single CEBPA mutations, define a subgroup of acute myeloid leukemia with a distinctive gene expression profile that is uniquely associated with a favorable outcome, *Blood* 113(13), 3088-3091

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

Zhu J., Hastie T. (2004), Classification of gene microarrays by penalized logistic regression, *Biostatistics,* Volume 5(3), 427-443

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

Problem description

Background

Linear regression

Regression methods have become an integral component of any data analysis concerned with describing the relationship between a response variable and one or more explanatory variables. In any regression problem the key quantity is the mean value of the response variable, given the values of the explanatory variables. This quantity is expressed as $E[Y|x]$, where Y denotes the response variable and x denotes a value of the explanatory variable [1-3]. In linear regression we assume that this mean may be expressed as an equation linear in x.

$$E[Y|x] = \beta_0 + \beta_1 x \ (1)$$

This linear regression model is simple and often provides an adequate and interpretable description of how the inputs affect the output. We can also extend the model such that we can predict a real valued output $\mathbf{y}$ based upon the design matrix $\boldsymbol{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_\mathrm{p})$.

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon} \ (2)$$

The criterion of the fit normally used is the residual sum-of-squares (RSS). To minimize this criterion we must find the optimal vector $\widehat{\boldsymbol{\beta}}$ of regression coefficients.

$$RSS(\boldsymbol{\beta}) = (\boldsymbol{y} - \boldsymbol{X\beta})^{\boldsymbol{T}}(\boldsymbol{y} - \boldsymbol{X\beta}) \ (3)$$

It is easily shown that the estimates are given by:

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X^T X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \ (4)$$

Coefficient shrinkage

There are two reasons why we are often not satisfied with the least squares estimates (4).

- **Prediction accuracy:** The least squares estimates often have low bias but large variance. Prediction accuracy can sometimes be improved by shrinking some coefficients to zero. By doing so we sacrifice a little bit of bias to reduce the variance of the predicted values, and hence may improve the overall prediction accuracy.
- **Interpretation:** With a large number of predictors, we often would like to determine a smaller subset that exhibits the strongest effects. In order to get the "big picture", we are willing to sacrifice some of the small detail.

Ride regression

Ridge regression shrinks the regression coefficients by imposing a penalty on their size. The ridge coefficients minimize a penalized residual sum of squares:

$$\hat{\beta}^{ridge} = \underset{\beta}{argmax} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \;\; s.t \sum_{j=1}^{p} \beta_j^2 \leq s \;\; (5)$$

Equation (5) makes explicit the size constraint on the predictors. When there are many correlated variables in a linear regression model, their coefficients can become poorly determined and exhibit high variance. A wildly large positive coefficient on one variable can be canceled by a similarly large negative coefficient on its correlated cousin. By imposing a size constraint on the coefficients, this phenomenon is prevented from occurring. Here the regularization parameter s controls the amount of shrinkage: the smaller the value s, the greater the amount of shrinkage. This will result in the coefficients being shrunk to zero (and each other).

The lasso

The lasso is a shrinkage method like ridge, with subtle but important differences [4]. The lasso is a regularized estimation approach for regression models that constrains the L_1-norm of the regression coefficients.

$$\hat{\beta}^{lasso} = \underset{\beta}{argmax} \sum_{i=1}^{N} (y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 \;\; s.t \sum_{j=1}^{p} |\beta_j| \leq t \;\; (6)$$

The use of lasso as a regularization method has two main advantages. First advantage is that it shrinks the regression coefficients towards zero and automatically can set many of them exactly zero, depending on the magnitude of the regularization parameter t. This indicates that we apply variable selection somewhat analogous to forward or backward feature selection. This can be very useful for high-dimensional data, where the number of predictors is larger than the number of observations. Ultimately the variable selection will result in the obtainment of an interpretable prediction rule, and the shrinkage is also desirable to improve the prediction and to prevent overfit.

Logistic regression

Logistic regression is a model used for prediction of the probability of occurrence of an event by fitting data to a logistic curve. The logistic regression model is a generalized linear model that can be used to classify samples. Let us assume that we have dichotomous data, with classes $\{0, 1\}$, and the following definitions:

Odds:

- If an event has probability P, it has odds $\frac{P}{1-P}$
- Odds go from 0 to ∞

Assumptions:

- Response y_i Bernoulli distributed: P(y_i=1)=P
- Logistic regression: $\ln(\frac{P}{1-P}) = \eta_i$
- Linear predictor $\eta_i = \beta_0 + \vec{x}^t\vec{\beta}$

We can show that the model has the form:

$$P(\omega_0|X) = \frac{e^{\beta_0+\vec{x}^t\vec{\beta}}}{1+e^{\beta_0+\vec{x}^t\vec{\beta}}} \quad (7)$$
$$P(\omega_1|x) = \frac{1}{1+e^{\beta_0+\vec{x}^t\vec{\beta}}}$$

Logistic regression models are usually fit by maximum likelihood. The log-likelihood can be written as:

$$l(\beta) = \sum_{i=1}^{N}\left\{y_i\beta^T x_i - \log\left(1+e^{\beta^T x_i}\right)\right\} \quad (8)$$

An advantage is that this log-likelihood function is concave, lending itself optimally to the methods of Newton-Raphson. To find the optimal set of coefficients we need to calculate the gradient and the Hessian of the log-likelihood function. This can be easy done for dichotomous data, but is more elaborate for polytomous data.

Multinomial logistic regression

Logistic regression is most frequently employed to model the relationship between a dichotomous outcome variable and a set of covariates, but with a few modifications it may also be used when the outcome variable is polytomous [5]. Suppose the outcome variable Y takes a value in the unordered set $\{1, \cdots, g\}$. In the multinomial logistic regression model the probability of each outcome depends on the covariates $X_1, \cdots, X_p$ as

$$P(Y = s) = \frac{e^{\eta_s}}{\sum_{t=1}^{g} e^{\eta_t}} \quad (9)$$

where $\eta_s = \sum_{k=1}^{p} X_k\beta_{ks}$ is a linear function of the covariates. In this formulation of the model we have a regression coefficient β_{ks} for each combination of covariate k and outcome category s, and a separate vector of linear predictors η_s for each outcome category. The fitting of this model is somewhat more complex. Suppose we have samples outcomes $Y_1, \cdots, Y_n$, a corresponding $n \times (p+1)$ data matrix of covariates X and make the simplifying substitution $\mu_{is} = P(Y_i = s)$.

For notational convenience we write the y_{is}, μ_{is} and η_{is} in the form of long $ng \times 1$ vectors: $\boldsymbol{y} = (y_{11}, \cdots, y_{n1}, \cdots, y_{1g}, \cdots, y_{ng})^T$, $\boldsymbol{\mu} = (\mu_{11}, \cdots, \mu_{n1}, \cdots, \mu_{1g}, \cdots, \mu_{ng})^T$ and $\boldsymbol{\eta} = (\eta_{11}, \cdots, \eta_{n1}, \cdots, \eta_{1g}, \cdots, \eta_{ng})^T$. The linear predictors $\boldsymbol{\eta}$ are related to the vector of parameters $\boldsymbol{\beta} = (\beta_{11}, \cdots, \beta_{p1}, \cdots, \beta_{1g}, \cdots, \beta_{pg})^T$ through $\boldsymbol{\eta} = \boldsymbol{X\beta}$, where $\boldsymbol{X} = I_g \otimes X$,where $\otimes$ is the Kronecker product and I_g the $g \times g$ identity matrix.

The log-likelihood of the model is:

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n}\sum_{s=1}^{g} y_{is} \ln \mu_{is} \quad (10)$$

which has gradient $\frac{\partial l(\beta)}{\partial \beta} = \boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{\mu})$ and Hessian $\frac{\partial^2 l(\beta)}{\partial \beta^2} = -\boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}$. Where the $ng \times ng$ matrix $\boldsymbol{W}$ is given by:

$$\boldsymbol{W} = \begin{pmatrix} W^{11} & W^{12} & \cdots & W^{1g} \\ W^{21} & W^{22} & & \vdots \\ \vdots & & \ddots & \\ W^{g1} & \cdots & & W^{gg} \end{pmatrix}$$

where each $\boldsymbol{W^{ij}}$ is a diagonal matrix with

$$diag(\boldsymbol{W^{st}}) = diag(\boldsymbol{W}^{ts}) = \begin{cases} (-\mu_{\boldsymbol{is}}\mu_{\boldsymbol{it}}, \cdots, \mu_{ns}\mu_{nt})^T & if\ s \neq t \\ (\mu_{1s}(1-\mu_{1s}), \cdots, \mu_{ns}(1-\mu_{ns}))^T & if\ s = t \end{cases}$$

Due to the fact that the Hessian matrix is singular we use the Moore-Penrose inverse of the Hessian in the Newton-Raphson algorithm.

Problem definition

There are many different methods to perform multi-class classification but the use of multinomial logistic regression as classifier could have an additional advantage. We could namely integrate coefficient shrinkage in the classification, such that we not only improve the prediction accuracy but also perform variable selection. This thesis looks into state-of-the-art methods currently developed to integrate penalization into the estimation of the regression coefficients. The main objective is to use the method upon microarray data originating from different classes. In this thesis the following issues will be addressed:

Large Hessian: The optimal regularization parameters in the penalization functions can best be empirically determined by methods such as leave-one-out cross validation or the Aikake Information Criterion (AIC). This implies that the Newton-Raphson algorithm must be performed for a wide range of the regularization parameter until convergence. In the multinomial logistic regression model the Hessian can become quite large, resulting in unfeasible computation times. To resolve this issue we should investigate Quasi-Newton methods, such as the limited memory Newton-Raphson method (L-BFGS-B). The advantage is that it can integrate bounds in the optimization. Generally, these types of methods need more optimization steps, but need less computation time per step.

Integration penalization: One open problem of the multinomial logistic regression is that the capability of penalization has not yet been integrated into the model. An integration of the penalization could lead to more accurate class predictions and has the advantage of variable selection. During this thesis we would like to integrate to types of penalization: ride regression, lasso. A fact is that microarray data is very high-dimensional (currently around 55.000 measurements), and in this case variable selection is desirable to obtain an interpretable prediction rule.

Suppose that we have a multi-class classification problem with classes $\{1, 2, \cdots, g\}$. In this case we have a regression coefficient vector for each class, $\vec{\beta}_i \;\; i \in \{1,2, \cdots, g\}$ and can be rewritten for notational convenience as a matrix:

$$\begin{bmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1g} \\ \beta_{21} & \beta_{22} & & \vdots \\ \vdots & & \ddots & \\ \beta_{p1} & \cdots & & \beta_{pg} \end{bmatrix}$$

The integration of penalization is in this case more complex due to the fact that the regression coefficients for each response variable are now related (row), which is naturally also for each regression coefficient vector (column). To avoid this problem we can reparameterize the model and exploit the structure of the matrix. This reparameterization describes the same model, but with different parameters. A downside is that the penalization is variant for reparameterization and might define a different penalization-structure resulting in different solutions. A clever definition of the penalization structure should be invariant to parameterizations and is a topic addressed during this thesis. The use of a proper reparameterization could also enhance the interpretability of the regression coefficients (e.g. coefficient testing).

Additionally we would like to integrate the idea of the group lasso for logistic regression [6-7]. This method groups input variables in so called “factors”. In microarray experiments a natural grouping might for example be; genes that have a similar function or pathways. In this case we are trying to find regression coefficient vectors that best explain the output variable vector (11).

$$\vec{y} = \sum_{j=1}^{J} X_j \vec{\beta}_j + \vec{\varepsilon} \quad (11)$$

where $\vec{y}$ is an $n \times 1$ vector, $\varepsilon \sim N_n(0, \sigma^2 I)$, X_j is an $n \times p_j$ matrix corresponding to the *j*th factor and $\vec{\beta}_j$ is a coefficient vector of size p_j. The group lasso methodology does not try to discard individual predictors by penalization, but tries to discard entire groups of variables by trying to shrink some $\vec{\beta}_j$ to $\vec{0}$. This methodology has also been extended for logistic regression [7], but has not been integrated for the multi-class case. During this thesis we wish to integrate group lasso into the multinomial logistic regression model. In this case a pathway or group must be selectively be "switched" on or off for each class!

Constraints on parameters: Next to the constraints posed by the penalization terms on the regression coefficients, it is sometimes desirable to put additional constraints on the parameters. Unfortunately, there are not so many standard software packages which are able to put these constraints on the coefficients. We wish to investigate how to integrate these additional constraints in the penalized multinomial logistic regression model.

Goal and validation

The goal of this project is to develop an efficient algorithm to perform multi-class classification with the penalized multinomial logistic regression. Eventually this method shall also be validated on real data. The department of Pathology provided use with two datasets:

1. **Sporadic parathyroid carcinomas:** A total of 53 parathyroid tumors and 16 normal specimens of parathyroid tissue were obtained from the Leiden University Medical Center, Royal North Shore Hospital and Martin Luther University and can be roughly divided into 5 different classes based upon genetic aberrations [8]. These samples were profiled on spotted cDNA microarrays.
2. **Colorectal cancer:** A total of 79 samples were collected from patients treated from different hospitals dispersed over the Netherlands. The samples can be classified as being in subsequent tumor stages and are analyzed with spotted cDNA microarrays [9].
3. **Acute Myeloid Leukemia:** A total of 600 Acute Myeloid Leukemia samples were obtained from the Erasmus Medical Center in Rotterdam with the HGU133A 2.0 PLUS microarray with around 55.000 probe sets. These samples have been fully characterized, such as its karyotype and mutation status. Particular groups of samples have a genetic mutation leading to a poor prognosis were as other have a more favorable prognosis. It remains to this day still a question what the underlying pathogenesis is [11-12].

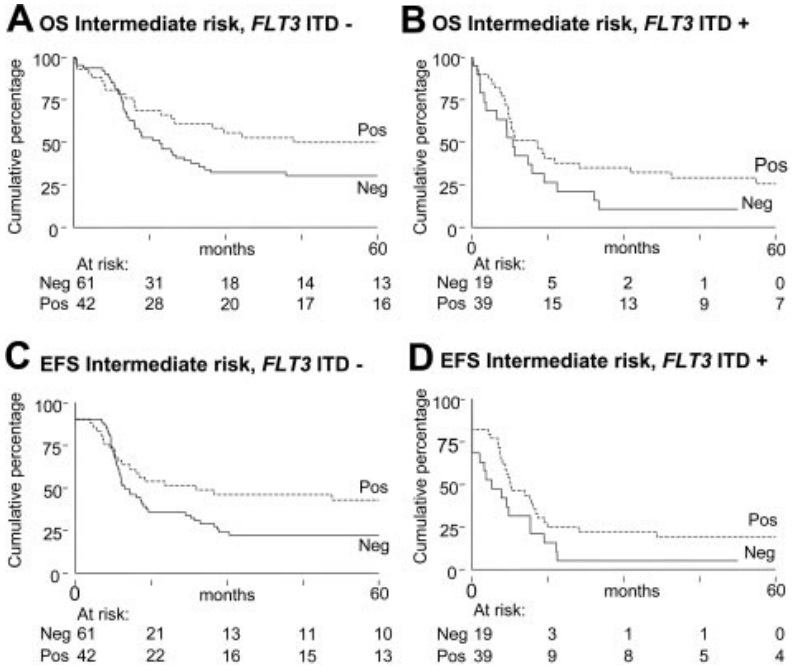
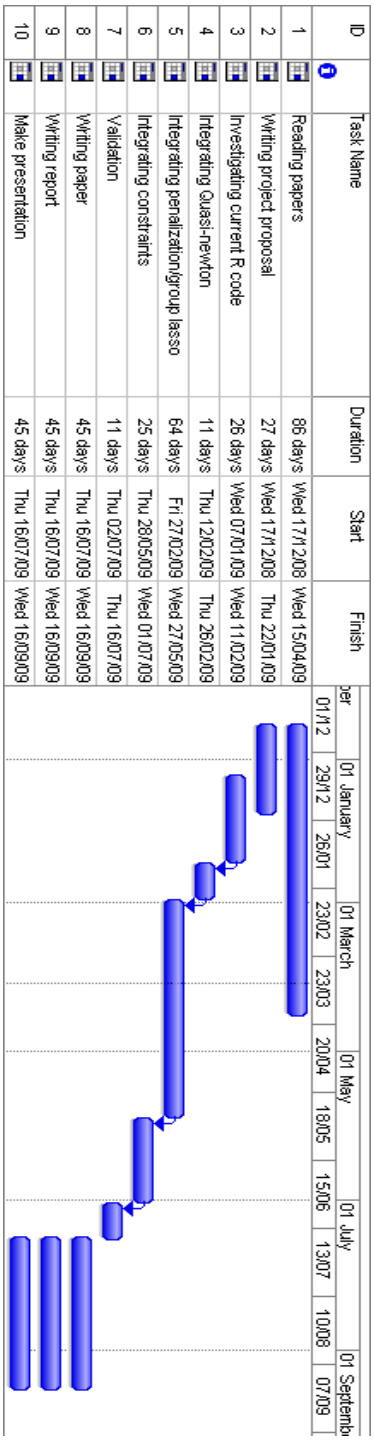


Figure 1: Kaplan-Meier curves for patients with an FLT3 ITD mutation and/or NPM1 mutation.

If need be, additional datasets can be retrieved from repositories, such as Gene Expression Omnibus [10]. There are multiple ways to general methods to validate the model, but we should also devise methods to infer the goodness-of-fit of the model [5]. This method represents a score test to infer the fit of the multinomial logistic regression model. Another point of validation could be to see if the selected covariates (e.g. genes) are biomarkers previously found associated, in literature, with the particular types of tumor.

Planning

To make sure that the project is not delayed or diverges from its original goal we give in this chapter a tight planning. This planning contains milestones, sub-goals and project boundaries such that the project progresses according plan.



ID	Task Name	Duration	Start	Finish
1	Reading papers	86 days	Wed 17/12/08	Wed 15/04/09
2	Writing project proposal	27 days	Wed 17/12/08	Thu 22/01/09
3	Investigating current R code	26 days	Wed 07/01/09	Wed 11/02/09
4	Integrating Quasi-newton	11 days	Thu 12/02/09	Thu 26/02/09
5	Integrating penalization/group lasso	64 days	Fri 27/02/09	Wed 27/05/09
6	Integrating constraints	25 days	Thu 28/05/09	Wed 01/07/09
7	Validation	11 days	Thu 02/07/09	Thu 16/07/09
8	Writing paper	45 days	Thu 16/07/09	Wed 16/09/09
9	Writing report	45 days	Thu 16/07/09	Wed 16/09/09
10	Make presentation	45 days	Thu 16/07/09	Wed 16/09/09

References

[1] Hastie T., Tibshirani R and Jerome Friedman (2001). *The elements of statistical learning.* Springer.

[2] Hosmer D.W. and Lemeshow S. (1998). *Applied logistic regression*. John Wiley and Sons Inc.

[3] Motulsky H. (1995). *Intuitive biostatistics.* Oxford University Press.

[4] Tibshirani R. (1996). *Regression shrinkage and selection via the LASSO.* Journal of the Royal Statistic Society series B-Methodology 58(1), 267-288

[5] Goeman J.J. and le Cessie S (2006). *A goodness-of-fit test for multinomial logistic regression*. Biometrics. 2006 Dec;62(4):980-5.

[6] Yuan M. and Lin Y (2006). *Model selection and estimation in regression with grouped variables*. Journal of the Royal Statistic Society series B-Methodology 68(1), 49-67.

[7] Meier L., van de Geer M. *and* Bühlmann P (2008). *The group lasso for logistic regression*. Journal of the Royal Statistic Society series B-Methodology 70(1), 53-71.

[8] Haven C.J., Howell V.M., Eilers P.H.C., et al (2004). *Gene Expression of Parathyroid Tumors: Molecular Subclassification and Identification of the Potential Malignant Phenotype.* Cancer research 64, 7405-7411.

[9] Lips E.H., van Eijk R., et al. (Submitted). *Integrating chromosomal aberrations and gene expression profiles to dissect rectal cancer.*

[10] Gene Expression Omnibus. [http://www.ncbi.nlm.nih.gov/geo/]

[11] Valk PJM, Verhaak RGM, Beijen MA, *et al.* (2004), *Prognostically useful gene-expression profiles in acute myeloid leukemia,* New England Journal of Medicine Volume 350:1617-1628

[12] Verhaak RGW, Goudswaard RW, van Putten W, Bijl MA, Sanders MA, et al. (2005), *Mutations in nucleophosmin (NPM1) in acute myeloid leukemia (AML): association with other gene abnormalities and previously established gene expression signatures and their favorable prognostic significance,* Vol. 106, No. 12, pp. 3747-3754