PhD thesis submitted at the
University of Copenhagen
Faculty of Science

Title: High dimensional multiclass classification with applications to cancer diagnosis

Martin Vincent

May 31, 2013

Abstract

Probabilistic classifiers are introduced and it is shown that the only regular linear probabilistic classifier with convex risk is multinomial regression. Penalized empirical risk minimization is introduced and used to construct supervised learning methods for probabilistic classifiers. A sparse group lasso penalized approach to high dimensional multinomial classification is presented. On different real data examples it is found that this approach clearly outperforms multinomial lasso in terms of error rate and features included in the model. An efficient coordinate descent algorithm is developed and the convergence is established. This algorithm is implemented in the `msgl` R package.

Examples of high dimensional multiclass problems are studied, in particular examples of multiclass classification based on gene expression measurements. One such example is the – clinically important – problem of identifying the primary tumor site of lever metastases, this particular problem is studied in detail. In order to adjust for the lever contamination found in biopsies of metastases a computational contamination model is develop. The contamination model is presented in a domain adaption framework and a simulation based domain adaption strategy is presented. It is shown that the presented computational contamination approach drastically improves the primary tumor site classification of lever contaminated biopsies of metastases. A final classifier for identification of the primary tumor site is developed. This classifier is validated on an independent validation set consisting of lever biopsies of metastases with varying tumor content.

Resumé

Stokastiske klassifikatore introduceres og det bliver vist at den eneste regulære linear stokastiske klassifikator med konvex risiko er multinomial regression. Straffet empirisk risiko minimering introduceres og bruges til at konstruere supervisered lærings metoder for stokastiske klassifikatore. En sparse group lasso straffet tilgang til høj dimensional multinomial klassifikation præsenteres. På forskellige data eksempler ses det tydeligt at denne tilgang inducere bedre modeler end multinomial lasso. En koordinatvis optimerings algoritme udvikles og konvergensen af denne vises. Denne algoritme er implementeret i R pakken `msgl`.

Eksempler på høj dimensional mange klasse klassifikations problemer undersøges, specielt genekspression eksempler. Et sådanne eksempel er – det kliniske vigtige – problem med identifikation af den primære tumor af lever metastaser, dette problem studieres i detaljer. En stokastisk kontaminerings model udvikles for at juster for den lever kontaminering der findes i biopsier af lever metastaser. Og det vises at denne model forbedre klassifikationen af den primære tumor. En endelig klassifikator udvikles og valideres på et uafhængigt sæt af lever biopsier af metastaser.

Contents

Chapter 1

Introduction

This thesis consist of three introductory chapters, the three primary articles Vincent and Hansen [21], Vincent et al. [22], Perell et al. [12], one software package Vincent [20] and one secondary article Søkilde et al. [14]. The status of this work is disclosed in section 1.1 below. The main theme of the thesis is high dimensional multiclass classification, and a central application is identification of the primary tumor site of metastases. Secondary themes are domain adaption and non differentiable convex optimization.

We will approach high dimensional multiclass classification by a regularization approach, specifically by penalized empirical risk minimization. Where a classification problem is said to be *high dimensional multiclass* if there are three or more classes and a high number of covariates relative to the number of samples. We adapt the view that classification is estimation of the conditional probability of a class given the observed covariates, this viewpoint is termed *probabilistic discriminative models* by Bishop [2]. It is different from the – perhaps more traditional – view that classification is the problem of discriminating between classes.

Examples of discriminating classification methods are support vector machines for classification Vapnik [19], k-nearest neighbor algorithm and classification trees. Examples of probabilistic discriminative models are multinomial regression (sometimes also called multiclass logistic regression) and probit regression (see fore example Bishop [2]). Classifiers like linear discriminant analysis (LDA) models the joint density of the class and covariates, which implies that the conditional probability of a class is estimated. In fact, for LDA the conditional probability has the same form as in multinomial regression Hastie et al. [7] – but estimated differently. An advantage of the probabilistic notion is that not only one class is estimated but the probability of each of the classes.

Penalized empirical risk minimization is a penalized version of empirical risk minimization Vapnik [19]. A natural approach is to minimize the penalized log-likelihood risk of the probabilistic model, this corresponds to maximum likelihood estimation. In order for such methods to be well behaved convexity of the empirical risk is preferable. And as we shall see the only probabilistic model with convex log-likelihood empirical risk is – in broad terms – the multinomial regression model, this statement will be made precise in Chapter 2. Hence we are naturally lead to the multinomial regression model.

Empirical risk minimization is likely to produce overfitted solutions for high dimensional problems, we therefore use a penalty to regularize the solution. The penalty is essential in determining the characteristics of the resulting method, non-differentiable penalties will – if carefully chosen – induce model selection properties. This has been known for some time, with the primary example being the *lasso* or ℓ_1-norm penalty. As we shall see the ℓ_1-norm may

Data set	Classes	Covariates	Parameters	Samples
Primary Cancers	11	384	4k	199
Brain Tumor	5	7k	35k	40
Childhood Leukemia	4	8k	32k	60
Amazon Reviews	50	10k	500k	1500

Table 1.1: Data sets used in chapter 2 – 4. The table lists the number of classes, covariates, parameters for a multinomial model and samples.

not be the best possible penalty for multiclass classification, a better choice may be a sum of ℓ_2-norms[1] – i.e. a *group lasso* – penalty. Another attractive penalty is the combined ℓ_1-norm and ℓ_2-norm, this penalty is called the *sparse group lasso*.

Due to the non-differentiability of the above penalties the development of algorithms for solving such penalized optimization problems are not simple. The simplest case is ℓ_1-norm penalization which can be solved using coordinate descent. Efficient algorithms are available for solving these types of problems. The ℓ_2-norm penalty is slightly more complex, but can be solved using block coordinate descent. Algorithms for solving the sparse group lasso is somewhat more complex, we present and establish convergence of one such algorithm in Vincent and Hansen [21].

In order to compare, assess and select models some vocabulary is needed, Chapter 3 is primarily devoted to defining and illustrating such a vocabulary. A notion of *model characteristic* will be introduced, this abstraction broadens the notion of generalization error. The number of covariates include in the model is an example of a model characteristic. And as we shall see different penalties will produce methods with different relations between different model characteristics, as for example the expected generalization error and the expected number of covariates in the model.

We define the notion of *supervised learning method* and *parametrized supervised learning method*. Empirical risk minimization is an example of a supervised learning method and penalized empirical risk minimization an example of a parametrized supervised learning method – these examples will be introduced in Chapter 3. Parametrized supervised learners differ from supervised learns as they produce not a single model but a sequence of models, usually parametrized by a positive scaler. For penalized empirical risk minimization this scalar is the amount of regularization, that is the weight of the penalty. We will discuss various ways of comparing such methods and briefly model selection, that is how do we choose an optimal model among the produced sequence of models.

In the last part of Chapter 3 we will introduce a general notion of *subsampling procedures* of which cross validation and standard subsampling is an example. We will show that all such procedures induce unbiased estimates of the expected model characteristic. Moreover we derive a formula for the variance, and use this to briefly investigate the variance of cross validation and standard subsampling.

In Chapter 4 we study the solutions of the optimization problems associated with penalized empirical risk minimization. A good grasp of the solutions to these problems is a – in my opinion essential – first step towards a better understanding of the statistical properties of the estimators induced by penalized empirical risk minimization. Using standard convex analysis we derive a optimality condition of the solutions of optimization problems with sublinear penalties.

Sublinear penalties will be introduced and defined, they generalize the lasso, group lasso and

[1] Do not confusion the ℓ_2-norm with ridge regression, the ridge regression penalty is the square of the ℓ_2-norm.

Work	Journal / Repository	Status
Vincent and Hansen [21]	Computational Statistics & Data Analysis	under revision
Vincent et al. [22]	Bioinformatics	submitted
Vincent [20]	CRAN	on CRAN
Perell et al. [12]	Clinical Cancer Research	submitted
Søkilde et al. [14]		submitting June

Table 1.2: Status overview

sparse group lasso penalties. Furthermore we give an exact solution when the risk is quadratic and show how this can be used to get some geometric insight into the solution of such problems. This exact solution can also be used to derive generic algorithms for solving sublinear penalized empirical risk minimization problems.

Throughout the thesis various real data sets will be used to illustrate concepts and methods, the characteristic of the data sets used in the introductory chapters is list in Table 1.1. Short descriptions and references related to the data sets can be found in Appendix C.

1.1 Status and comments

The primary contribution, which we develop in Vincent and Hansen [21], is sparse group lasso multinomial regression. The idea is to group parameters corresponding to the same covariate and then estimate models using the sparse group lasso penalized maximum likelihood estimator. We show that for many real data sets this sparse group lasso estimator is superior to a plain lasso estimator in terms of archiving a lower error rate with fewer covariates included in the model. Another important contribution, made in Vincent and Hansen [21], is the development of a coordinate descent algorithm – suitable for high dimensional problems – for solving the sparse group lasso penalized maximum likelihood estimates.

The algorithm developed in Vincent and Hansen [21] was implemented in `C++`. This implementation was used as a basis for the `msgl` R package Vincent [20]. The `msgl` package is available on CRAN.

In Vincent et al. [22] we develop a contamination model and use it to improve the identification of the primary tumor site of lever metastases. The contamination model is presented in a domain adaption framework and we suggest a simulation based domain adaption strategy. We use the contamination model in combination with the presented domain adaption strategy, and show that the combined approach drastically improves the primary tumor site classification of lever contaminated biopsies of metastases.

A clinically applicable classifier for identification of the primary tumor site of lever metastases is developed and validated in Perell et al. [12]. A long range of classifiers for identification of the primary tumor site have been published see [2-7] in the reference list of Perell et al. [12]. The developed classifier is designed to be clinically applicably and is – to our knowledge – the first which is systematically validated on core biopsies of metastases – contrary to many of the other classifiers developed. It is developed using the contamination model of Vincent et al. [22] and it can cope with high levels of lever contamination.

Chapter 2 is for the most part a summery of standard theory, however with a focus on convexity of the risk not normally seen. Theorem 2 states that the only regular linear model with convex log-likelihood risk is the multinomial model, this result is to my knowledge a new

result. Chapter 3 primarily introduce and illustrates a vocabulary suitable for use with penalized empirical risk minimization for high dimensional problems. At the end of Chapter 3 a general notion of subsampling procedures – of which cross validation is an example – is presented and it is shown that all such procedures induce unbiased estimators of the expected generalization error. Moreover a formula – Proposition 3 – for the different components of the variance of these estimators is established. Proposition 3 can quite easily be established, however, I am not aware of any other clear statements of this result. The results of Chapter 4 is to my knowledge new, it is work in progress.

The status of work intended for publication of this thesis is listed in table 1.2. The manuscript Søkilde et al. [14] is placed in Appendix D, it is work carried out in the year 2011 at Exiqon, it is a first attempt at constructing a classifier for identification of primary tumor site. The attached manuscript was submitted – but rejected – to Journal of Molecular Biology. A new manuscript is being prepared and the corresponding author (R. Søkilde) have informed me that the revised manuscript will be submitted in June 2013.

Chapter 2

Classification models

2.1 Introduction

In this chapter the classification problem will be formalized. There are two ways to approach classification, one view is that classification is separation between groups. Another view is that classification is estimation of the conditional probability of the class given the observed covariates. In this thesis we take the later viewpoint, that is classification is not just specification of the class but the probability of the classes.

The error of a classifier is measured using a *loss* function, we introduce this concept and show that some regularity of the loss function is required in order to ensure that the *Bayes classifier* is optimal. The 01 loss and the log-likelihood loss are introduced and shown to be regular in the sense that the Bayes classifier has the optimal *generalization error*.

We introduce parametric models for classification and define the *risk* and *empirical risk* of a classifier. The empirical risk may be seen as an approximation of the risk. For high dimensional problems this approximation may have defects which can lead to overfitting – we will see an illustration of this.

Convexity is a desirable property for the risk and empirical risk, and we will discuss the relation between the convexity of the risk and the convexity of the loss and the model. We will show that convexity of the loss function, composed with the model, is necessary and sufficient in order to achieve convexity of the risk and empirical risk. We shall in particular see that the multinomial regression model is – in broad terms – the only linear model with convex log-likelihood risk – in section 2.4 we will make this precise.

A discussion of quadratic approximations of the empirical risk have been placed in appendix A.1. Identifiability is discussed briefly in appendix A.2 and in Vincent and Hansen [21].

2.2 The classifier and the loss

A $p \in \mathbb{N}$ dimensional classification problem with $K \in \mathbb{N}$ classes consist of a random vector $X \in \mathbb{R}^p$ and a discrete random variable Y taking K different values. The vector X is called the covariate vector and the variable Y the response or class variable. The joint distribution of (X, Y) will be denote by F and the joint density by f. The classes, that is the values of the response variable Y, is denoted by $\mathcal{S}_K \stackrel{\text{def}}{=} \{1, \ldots, K\}$. The number of classes is K and the *number of covariates* is p.

The classification problem is well defined if the class Y depend on the covariate vector X; prediction of the class Y given the covariates X is called *classification* in the statistical literature, see for example Wasserman [23], Hastie et al. [7]. In the computer science literature this problem is sometimes referred to as *pattern recognition*.

In the case of classification we seek to predict Y knowing X. A rule that predicts Y given X is called a *classifier*, this notion may be formalized by defining a classifier as a function

$$h : \mathbb{R}^p \to \mathcal{S}_K, \tag{2.1}$$

where for a given covariate vector $x \in \mathbb{R}^p$ the predicted class is $h(x)$. This viewpoint is often refereed to as pattern recognition Vapnik [19] and h is called a *discriminant function* Bishop [2], it is a formalization of the view that classification is the problem of discriminating between groups.

Another approach to classification is to estimate the probabilities of the classes given the observed covariates. More precisely the problem of estimating the conditional density of Y given X, this notion is termed *Probabilistic Discriminative Models* by Bishop [2]. The advantage of the probabilistic viewpoint over the discriminative is that additional information about the underlying conditional probability is estimated. In this thesis we will adapt the probabilistic view, hence; a classifier is a rule that to each $x \in \mathbb{R}^p$ assigns a probability distribution on the finite set with K elements. The set of all such probability distributions is the simplex

$$\Delta^K \stackrel{\text{def}}{=} \left\{ p \in [0,1]^K \mid \sum_{i=1}^{K} p_i = 1 \right\}.$$

Therefore a precise definition, in accordance with the probabilistic viewpoint, of a classifier is:

Definition 1 (Classifier)**.** *A classifier is a function* $\mathfrak{p} : \mathbb{R}^p \to \Delta^K$.

If $\mathfrak{p}$ is a classifier and x an observed covariate, i.e a realization of X, then the k'th coordinate $\mathfrak{p}_k(x)$ of $\mathfrak{p}(x)$ is interpreted as the estimated conditional probability that $Y = k$ given $X = x$. In other words $\mathfrak{p}_k(x)$ is the estimated probability that the observation, x, belongs to class k.

Given a classifier h, as defined by (2.1) we may construct a (not very informative) classifier $\mathfrak{p}$ by assigning

$$\mathfrak{p}_k(x) = \begin{cases} 1 & k = h(x) \\ 0 & \text{otherwise} \end{cases}$$

for $x \in \mathbb{R}^p$. Given a classifier $\mathfrak{p}$ we may construct a classifier h, by choosing

$$h(x) \in \underset{k=1,\ldots,K}{\arg\max}\, \mathfrak{p}_k(x).$$

When viewing classification as discrete density estimation, it is clear that the true conditional density of Y given X is the theoretical optimal classifier. This classifier is called the Bayes classifier. The precise definition is:

Definition 2 (Bayes classifier)**.** *The Bayes classifier is the classifier* $\mathfrak{p}^{Bayes}$ *defined by*

$$\mathfrak{p}_k^{Bayes}(x) \stackrel{def}{=} P(Y = k | X = x).$$

In terms of the joint density f of (X, Y) the Bayes classifier

$$\mathfrak{p}_k^{\text{Bayes}}(x) = \frac{f(x,k)}{f_Y(k)}$$

where $f_Y(k) = \int f(x,k)\,\mathrm{d}x$ is the marginal density for Y. In practice we almost never have any prior knowledge about the joint distribution of (X,Y), in particular not the conditional distribution of Y given X. In other words we do not have access to the Bayes classifier, and we must therefore estimate the conditional distribution of Y given X, i.e. attempt to approximate the Bayes classifier.

2.2.1 The loss function

To compare and assess the performance of classifiers a measure of the error, or loss, of a prediction is needed. This error is measured by a *loss function*, where we shall take a loss function to be any function of the form

$$L : \Delta^K \times \mathcal{S}_K \to \mathbb{R} \cup \{\pm\infty\}.$$

Given an observation (x,y), i.e. a realization of (X,Y), the loss $L(\mathfrak{p}(x),y)$ can be interpreted as the error the classifier $\mathfrak{p}$ makes on the observation (x,y). For example, given two classifiers $\mathfrak{p}_1$ and $\mathfrak{p}_2$ the relation

$$L(\mathfrak{p}_1(x),y) \leq L(\mathfrak{p}_2(x),y)$$

may be interpreted as: classifier $\mathfrak{p}_1$ did a better job predicting the class than $\mathfrak{p}_2$ on the observation (x,y). In particular the 01 loss and the log-likelihood loss is of interest to us.

The *01 loss* measures if we got the class with the highest probability right. The loss is zero if the class with highest probability equals the observed class and one otherwise. When making this definition precise we must ensure that we define a function, that is we need to ensure that in the cases several classes have the same probability the value of the loss is uniquely determined. We therefore define the 01 loss by

$$L(p,y) = \begin{cases} 0 & \text{if } y \text{ equals the smallest index in } \underset{i=1,\ldots,K}{\arg\max}\, p_i \\ 1 & \text{otherwise} \end{cases}.$$

The *log-likelihood loss* is defined by

$$L(p,y) = -\log p_y.$$

As will become clear, see Lemma 1, the log-likelihood loss is, up to a constant, the log likelihood.

Let L be any loss function. Then the expected loss of a classifier is called the *generalization error*, risk, or true error. Note that the term generalization error does not specify the loss function.

Definition 3 (Generalization error)**.** *The generalization error of a classifier* $\mathfrak{p}$ *is*

$$\mathrm{err}(\mathfrak{p}) \overset{def}{=} \mathrm{E}\, L(\mathfrak{p}(X),Y).$$

Optimality of the Bayes classifier

Not all loss function will induce a sensible generalization error. Any sensible measure of error should be consistent with the optimality of the Bayes classifier, that is the optimality of the true conditional density. This implies that the Bayes classifier should be a minimizer of the generalization error induced by the loss function. In other words the Bayes classifier should have the lowest generalization error, and this requirement places restrictions on the loss function. Theorem 1 below gives a necessary and sufficient condition, on the loss function, that ensures optimality of the Bays classifier.

Theorem 1 (Optimality of the Bayes classifier)**.** *The following are equivalent*

(a) For any classifier $\mathfrak{p}$ *and any joint distribution of* (X, Y)

$$\mathrm{err}(\mathfrak{p}^{bays}) \leq \mathrm{err}(\mathfrak{p})$$

(b) For any $\pi \in \Delta^K$*,* π *is a minimizer of*

$$g(s) \stackrel{def}{=} \sum_{y=1}^{K} \pi_y L(s, y).$$

Proof. We shall prove the following implications

$$(a) \stackrel{1}{\Longrightarrow} (b) \stackrel{2}{\Longrightarrow} (a).$$

1. Let $\pi \in \Delta^K$, denote by $f_X(x)$ the marginal density of X and consider the joint density

   $$f(x, y) = \pi_y f_X(x).$$

   By condition (a) there exists an $x \in \mathbb{R}^p$ such that

   $$\mathrm{E}(L(\mathfrak{p}^{\mathrm{bays}}(X), Y) \mid X = x) \leq \mathrm{E}(L(\mathfrak{p}(X), Y) \mid X = x).$$

   Since

   $$\begin{aligned} \mathrm{E}(L(\mathfrak{p}(X), Y) \mid X = x) &= \sum_{y=i}^{K} L(\mathfrak{p}(x), y) P(Y = y \mid X = x) \\ &= \sum_{y=1}^{K} \pi_y L(\mathfrak{p}(x), y), \end{aligned}$$

   it follows that $\pi = \mathfrak{p}^{\mathrm{bays}}(x)$ is a minimizer of g.

2. Let $x \in \mathbb{R}^p$ and $\pi_y = P(Y = y \mid X = x)$, by (b) it follows that $\mathfrak{p}^{\mathrm{Bayes}}(x)$ is a minimizer of g. This implies that, for any classifier $\mathfrak{p}$

   $$\mathrm{E}(L(\mathfrak{p}^{\mathrm{Bayes}}(X), Y) \mid X = x) \leq \mathrm{E}(L(\mathfrak{p}(X), Y) \mid X = x),$$

   which in turn implies (a).

□

A loss function is said to be a *regular loss* if it comply with condition (b) of Theorem 1. The 01 loss and the log-likelihood loss are regular, see example 1 and 2 below. The generalization error of the Bayes classifier is called the *Bayes rate*. For a regular loss this is by Theorem 1 the lowest achievable generalization error.

A corollary to Theorem 1 is that for a regular loss the Bayes classifier is not only globally optimal but locally optimal. The precise statement is:

Corollary 1 (Conditional optimality of Bayes classifier)**.** *For any classifier* $\mathfrak{p}$ *and any joint distribution on* (X, Y)

$$\mathrm{E}(L(\mathfrak{p}^{bays}(X), Y) \mid X = x) \leq \mathrm{E}(L(\mathfrak{p}(X), Y) \mid X = x) \quad \textit{for all } x \in \mathbb{R}^p.$$

Example 1 (Regularity of log likelihood loss)**.** *We need to find the minimizer of*

$$g(s) = \sum_{y=1}^{K} -\pi_y \log s_y$$

subject to $\sum_{y=1}^{K} s_y = 1$*. The Lagrangian is*

$$L(s, \lambda) = \sum_{y=1}^{K} -\pi_y \log s_y + \lambda \sum_{y=1}^{K} s_y.$$

Since g *is convex the minimizer* s^* *is a solution to the equations*

$$-\frac{\pi_y}{s_y} + \lambda = 0 \ \text{ for } y \in 1, \ldots, K$$

and $\sum_{y=1}^{K} s_y = 1$*. This implies that* $s = \pi$*.*

Example 2 (Regularity of 01 loss)**.** *We have that*

$$g(s) = \sum_{y=1}^{K} -\pi_y L(s, y) = 1 - \pi_i$$

where i *is the smallest index in* $\arg\max_{i=1,\ldots,K} s_i$*. It follows that*

$$g(\pi) = 1 - \max\{\pi_1, \ldots, \pi_K\} \leq g(s).$$

2.3 Parametric models

Let B denote a set of parameters, that is a subsets of an euclidean space.

Definition 4 (Parametric model)**.** *A parametric model for a classifier is a function*

$$\mathfrak{p} : B \times \mathbb{R}^p \to \Delta^K.$$

Given an parametric model, we say that a $\beta \in B$ is a model for the classifier $\mathfrak{p}(\beta)$. A model $\beta \in B$ induces a density on (X, Y) we denote this density by f_β. Since a parametric model $\mathfrak{p}$, as defined above, only models the conditional density of Y given X, it follows that the joint density f_β is

$$f_\beta(x, y) = \mathfrak{p}_y(\beta)(x) f_X(x) \tag{2.2}$$

where f_X is the marginal density for X.

When viewed as a function of the parameters $\beta \in B$ the generalization error is called the *risk* of the model β.

Definition 5 (Risk)**.** *The risk of the model* $\beta \in B$ *is*

$$R(\beta) \stackrel{def}{=} \mathrm{E}\, L(\mathfrak{p}(\beta)(X), Y).$$

A classifier may be obtained by minimization of the risk, this approach is called *risk minimization*. However since we do not know the distribution F of (X, Y) the risk is unknown, we need to estimate it from data. Before proceeding, we need to define what precisely we mean by data. A *random data set*, or *random sample* Cramér [5], is a collection of N samples drawn from the distribution F.

Definition 6 (Random data set)**.** *A* random data set *of size N is a random vector*

$$\mathfrak{D} = ((X_1, Y_1), \ldots, (X_N, Y_N))$$

where $(X_1, Y_1), \ldots, (X_N, Y_N)$ are i.i.d. according to the distribution F.

A realization of a random data set is a data set $D = ((x_1, y_1), \ldots, (x_N, y_N)) \in (\mathbb{R}^p \times \mathcal{S}_K)^N$.

2.3.1 The empirical risk

We may estimate the risk by the mean of a sample of losses. That is given a random data set $\mathfrak{D}$ we define:

Definition 7 (Empirical risk)**.** *The empirical risk of the model $\beta \in B$ is*

$$\hat{R}_{\mathfrak{D}}(\beta) \overset{def}{=} \frac{1}{N} \sum_{i=1}^{N} L(\mathfrak{p}(\beta)(X_i), Y_i)$$

If we by $\hat{F}_{\mathfrak{D}}$ denote the *empirical distribution* of $\mathfrak{D}$, or *the distribution of the sample* Cramér [5], then the empirical risk is the risk $R(\beta)$ with the joint distribution of (X, Y) being $\hat{F}_{\mathfrak{D}}$. The empirical risk, as defined above, is a random variable. For a realization D of $\mathfrak{D}$ we will also use the notation

$$\hat{R}_D(\beta) = \frac{1}{N} \sum_{i=1}^{N} L(\mathfrak{p}(\beta)(x_i), y_i).$$

Since we do not know the risk we seek instead an approximate minimizer of the risk by minimizing an approximation of the risk obtained from data, namely the empirical risk. This approach is called the *empirical risk minimization* principle, see Vapnik [19].

In practice empirical risk minimization may exhibit pronounced overfitting problems – as is often the case for high dimensional problems. In order to make this more precise assume given a realization D of the data set $\mathfrak{D}$. The Bayes classifier may not be a minimizer of the empirical risk $\hat{R}_D$. For high dimensional problems the Bayes classifier may be far from the minimizer of $\hat{R}_D$. In fact, for the 01 loss, we may often be able to construct a classifier $\tilde{\mathfrak{p}}$ with

$$\sum_{i=1}^{n} L(\tilde{\mathfrak{p}}(x_i), y_i) = 0,$$

that is with zero empirical risk. This classifier is constructed by requiring that

$$\tilde{\mathfrak{p}}(x_i)_k \approx I(k = y_i). \tag{2.3}$$

The classifier $\tilde{p}$ may not be in the set of parameterizable models, i.e. there may not exist a $\beta \in B$ such that $\tilde{\mathfrak{p}} = \mathfrak{p}(\beta)$. However when $N \ll p$ it is often possible to construct $\tilde{\mathfrak{p}}$ as a linear classifier. To avoid this overfitting problem we could, for example, use some from of regularization and thereby obtain a *regularized minimizer*. In Figure 2.1 we see that near the minimizer obtained by solving the underdetermined linear system (2.3) the empirical risk is a bad approximation of the risk (as estimated by a large test set). However near the regularized minimizer obtained by penalized risk minimization the empirical risk approximates the risk much better.

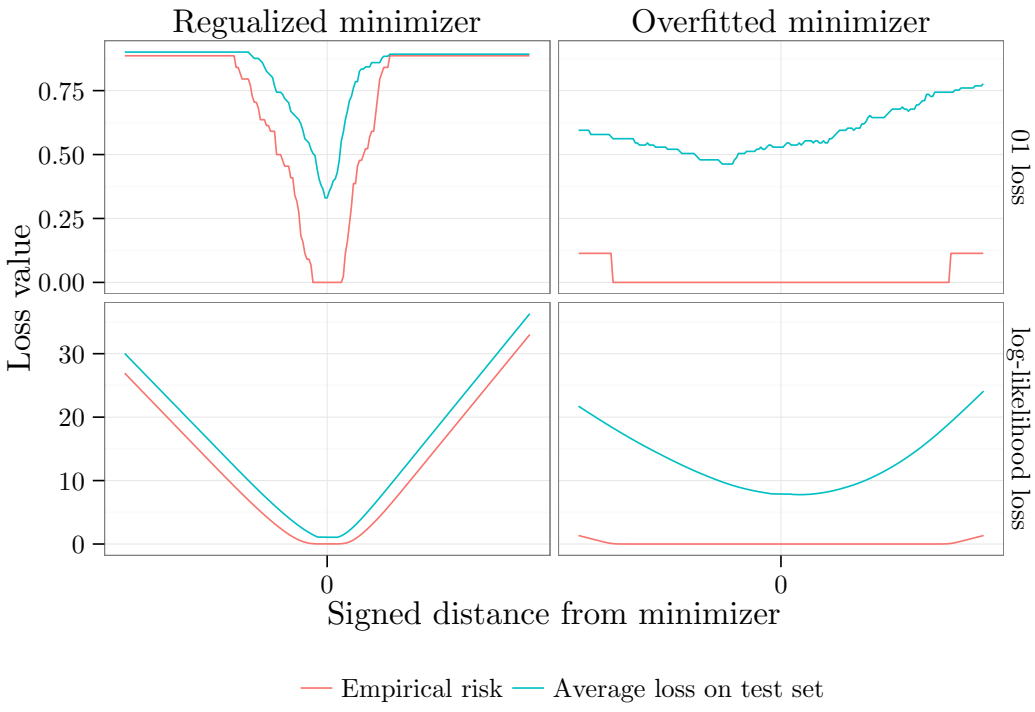


Figure 2.1: The value of the empirical risk, on a curve in the space of parameters B. The curve on the left intercept a regularized minimizer and the curve on the right intercept a overfitted minimizer. The empirical risk was constructed using 44 samples of the 11 class *Primary Cancers* data set – 5 in each class except for on class with 4 samples. The test set consisted of 121 samples, with 12-15 in each class. The regularized minimizer is obtained by penalized risk minimization using a group lasso . The overfitted minimizer is obtained by solving the K underdetermined linear systems (2.3).

2.3.2 Convexity of the risk

Convexity of the risk and empirical risk is a desirable property. Some of the reasons are:

- Every minimizer is global and the set of all minimizers is convex.
- The well established theory and technology of convex optimization can be applied in the development and construction of risk minimization algorithms.

For an introduction to convex optimization see Boyd and Vandenberghe [3]. By Proposition 1 below the convexity of the composition of the loss and the parametric model is equivalent to convexity of the risk and the empirical risk. Convexity of this composition is, in particular, necessary to ensure convexity of the empirical risk – regardless of the distribution on (X, Y). It is therefore an advantage to choose a loss function such that this composition is convex, otherwise we will have to deal with a non convex risk and empirical risk.

Proposition 1. *The following is equivalent*

(a) The function

$$\beta \to L(p(\beta)(x), y)$$

is convex for all $(x, y) \in \mathbb{R}^p \times \mathcal{S}_K$.

(b) For any distribution of (X, Y) *every realization of the empirical risk is convex.*

(c) For any distribution of (X, Y) *the risk is convex.*

Proof. Let, as usual, f denote the density of (X, Y). We shall prove

$$(a) \overset{1}{\Longrightarrow} (b) \overset{2}{\Longrightarrow} (c) \overset{3}{\Longrightarrow} (a).$$

1. Since a nonnegative weighted sum of convex functions is convex it follows that any realization of the empirical risk is convex.

2. Since any realization of the empirical risk is convex then for any $(x, y) \in \operatorname{supp}(f)$ the function $\beta \to L(p(\beta)(x), y)$ is convex. It follows that

$$\begin{aligned} R(\lambda\beta_1 + (1-\lambda)\beta_2) &= \sum_{y=1}^{K} \int_{\mathbb{R}^p} L(p(\lambda\beta_1 + (1-\lambda)\beta_2)(x), y) f(x, y) \, \mathrm{d}x \\ &\leq \sum_{y=1}^{K} \int_{\mathbb{R}^p} \left[\lambda L(p(\beta_1)(x), y) + (1-\lambda) L(p(\beta_2)(x), y)\right] f(x, y) \, \mathrm{d}x \\ &= \lambda R(\beta_1) + (1-\lambda) R(\beta_2) \end{aligned}$$

3. Given $(x, y) \in \mathbb{R}^p \times \mathcal{S}_K$ choose f degenerate at (x, y).

□

2.3.3 Relation to maximum likelihood

Lemma 1 below states the relation between the log-likelihood risk and the log-likelihood function.

Lemma 1. *For the log-likelihood loss it holds that*

$$\ell_{\mathfrak{D}}(\beta) = N\hat{R}_{\mathfrak{D}}(\beta) - \sum_{i=1}^{N} \log f_X(X_i)$$

where ℓ is the negative log-likelihood function.

Proof. The joint density of (X, Y) under the model is given by (2.2), the log likelihood is therefore

$$\begin{aligned}
\ell_{\mathfrak{D}}(\beta) &= \sum_{i=1}^{N} -\log f(X_i, Y_i) \\
&= \sum_{i=1}^{N} -\log \mathfrak{p}_{Y_i}(\beta)(X_i) - \log f_X(X_i) \\
&= N\hat{R}_{\mathfrak{D}}(\beta) - \sum_{i=1}^{N} \log f_X(X_i).
\end{aligned}$$

□

The lemma implies that minimizing the log likelihood risk is equivalent to minimizing the log likelihood. Risk minimization with the log-likelihood loss therefore amounts to maximum likelihood estimation. Hence maximum likelihood in the problem of classification may be viewed as a special case of risk minimization.

2.4 Linear models

For linear classification models the parameters are naturally organized in a matrix;

$$\beta = \underbrace{\left.\begin{pmatrix} \beta_{11} & \cdots & \beta_{1p} \\ \vdots & \ddots & \vdots \\ \beta_{K1} & \cdots & \beta_{Kp} \end{pmatrix}\right\}}_{\text{Covariates}} \text{Classes} \tag{2.4}$$

such that rows correspond to classes and columns to covariates.

Each $K \times p$ matrix β defines a map $\eta : \mathbb{R}^p \to \mathbb{R}^K$ by setting $\eta(x) \stackrel{\text{def}}{=} \beta x$ – if the parameter β needs to be explicitly stated then we will use the notation η_β instead of η. The elements of the vector $\eta(x)$ are called *linear predictors*, with the organization (2.4) the k'th element of η is the linear predictor corresponding class k. Denote the space of $K \times p$ real matrices by B, then a parametric model $p : B \times \mathbb{R}^p \to \Delta^K$ for a classifier is said to be linear if

$$p(\beta) = h \circ \eta_\beta$$

for a function $h : \mathbb{R}^K \to \Delta^K$. A linear model is specified completely by specifying the function h, we shall therefore say that h is the linear model. As usual we may add an intercept parameter to the model by considering the $p + 1$ dimensional problem where x is replaced by $(1, x)$.

A natural requirement for a linear model is that the conditional probability that $Y = k$ given X essentially only depends on the k'th linear predictor. We shall say that a linear model satisfying this requirement is *regular*, the precise definition is:

Definition 8. *A linear model is said to be regular if*

$$P(Y = k|X = x) = C(\eta)g(\eta_k)$$

for some functions $g : \mathbb{R} \to \mathbb{R}_+$ and $C : \mathbb{R}^K \to \mathbb{R}$.

Regularity of a linear model implies that the fraction

$$\frac{P(Y = k|X = x)}{P(Y = l|X = x)} = \frac{g(\eta_k)}{g(\eta_l)} \tag{2.5}$$

only depends on the k'th and l'th linear predictor. A regular linear model h has a special form, as lemma 2 below shows.

Lemma 2. *The linear model h is regular if and only if*

$$h(\eta) = \frac{1}{\sum_{i=1}^K g(\eta_i)}(g(\eta_1), \ldots, g(\eta_K)).$$

for some function $g : \mathbb{R} \to \mathbb{R}_+$.

Proof.

$$1 = \sum_{k=1}^K P(Y = k|X = x) = C(\eta) \sum_{k=1}^K g(\eta_k).$$

□

A regular linear model may not be identifiable, this is however not a problem since parameters may be interpreted though equation (2.5). This is in particular the case for the (symmetric) multinomial regression model, for a discussion see Vincent and Hansen [21] and appendix A.2 – the multinomial model will be introduced below.

2.4.1 Decision boundaries

The decision boundary between class k and class l is the set

$$\{x \in \mathbb{R}^p \mid P(Y = k|X = x) = P(Y = l|X = x)\}.$$

For a regular linear model the decision boundary is determined by the equation $g(\eta_k) = g(\eta_l)$. Hence, if g is injective then the decision boundary is the hyperplane given by

$$\eta_k - \eta_l = 0.$$

2.4.2 The multinomial model

Multinomial regression is the regular linear model with $g = \exp$. In this section we show that the multinomial regression model is the only regular linear model fulfilling the following three requirements.

1. The function g defining the linear model h is twice continues differentiable.
2. The function g is unbounded.
3. For any distribution on (X, Y) the log-likelihood risk is convex.

This implies that if we want to specify another regular linear model then at least one of the above requirements would have to be given up. We could for example replace requirement 3 by

$$\text{for any distribution on } (X, Y) \text{ the } L\text{-risk is convex,}$$

if the risk is convex for another loss function L.

The main theorem of this section is:

Theorem 2. *The only regular linear model fulfilling the three requirements above is the multinomial regression model.*

Proof. The statement follows by lemma 1 and 3 and by noting that composition with an affine function preserves convexity. □

Lemma 3. *Let h be a regular linear model and assume that $g : \mathbb{R} \to \mathbb{R}_+$ is twice continues differentiable and unbounded. Then the function*

$$L_k(\eta) = -\log h_k(\eta)$$

is convex for every $k \in \mathcal{S}_K$ if and only if $g(s) = c_1 \exp(c_2 s)$ with $c_1 > 0, c_2 \neq 0$.

Proof. Let $g(s) = c_1 \exp(c_2 s)$ with $c_1 > 0, c_2 \neq 0$. It follows that L_k is convex, since the log-sum-exp $\eta \to \log \sum_{k=1}^{K} \exp(\eta_k)$ is convex. For the converse, assume that L_1 is convex. Below we will show that this implies that

$$g'(s)^2 - g''(s)g(s) = 0. \tag{2.6}$$

And since the second order differential equation (2.6) has the solutions $g(s) = c_1 \exp(c_2 x)$, with constants $c_1 > 0, c_2 \neq 0$, the claim follows.

We still need to show (2.6), in order to do this note that the convexity of L_1 implies that the function

$$\mathbb{R} \ni s \to -\log \frac{g(s)}{g(s) + c} \tag{2.7}$$

is convex for every $c > 0$. The second derivative of (2.7) is

$$g''(s) \left(\frac{1}{g(s) + c} - \frac{1}{g(s)} \right) + g'(s)^2 \left(\frac{1}{g(s)^2} - \frac{1}{(g(s) + c)^2} \right). \tag{2.8}$$

Convexity of (2.7) implies that the second derivative (2.8) is nonnegative, it follows that

$$g'(s)^2 \left(1 - \left(\frac{g(s)}{g(s) + c} \right)^2 \right) - g''(s)g(s) \left(1 - \frac{g(s)}{g(s) + c} \right) \geq 0.$$

We must therefore have that

$$g'(s)^2\left(1+\frac{g(s)}{g(s)+c}\right)-g''(s)g(s)\geq 0,$$

and since this has to hold for every $c>0$ it implies that

$$g'(s)^2-g(s)g''(s)\geq 0.$$

The convexity of L_1 also implies that the function

$$\mathbb{R}\ni s\to -\log\frac{c_1}{c_1+c_2+g(s)} \tag{2.9}$$

is convex for every $c_1,c_2>0$. By using that the second derivative of (2.9) is nonnegative we find that

$$\left(\frac{g(s)}{g(s)+c_1+c_2}\right)g'(s)^2-g(s)g''(s)\leq 0$$

and since this has to hold for all constants $c_1,c_2>0$ it follows that

$$g'(s)^2-g(s)g''(s)\leq 0.$$

□

2.5 An empirical model

In Chapter 3 we will use simulated data to illustrate various concepts. To do this we will obtain a model β_0, for example using the multinomial group lasso estimator, on a template data set. Samples is then simulated from a empirical distribution F_{sim} corresponding to the estimated model β_0, that is the density of F_{sim} is given by (2.10). The advantage of this *empirical model* is that we know the distribution of the data, and this allow us to compute the Bayes rate, the true error and the variance of the loss.

The simulation scheme uses a real data set

$$D=\{(x_1,y_1),\ldots,(x_N,y_N)\}$$

as a *template* for the simulation. The simulation procedure is:

1. The template data set D is randomly split into two disjoint parts D_1 and D_2 containing respectively N_1 and N_2 samples.
2. A parametric model β_0 is estimated using D_1 as training data.
3. Let $f_{\hat{X}}$ denote the density of the empirical distribution of X in D_2. Draw N samples from the distribution F_{sim} with density

$$f_{\text{sim}}(x,y)\overset{\text{def}}{=}\mathfrak{p}_y(\beta_0)(x)f_{\hat{X}}(x). \tag{2.10}$$

These N samples constitute the simulated data set.

Clearly the Bayes classifier $\mathfrak{p}^{\text{Bayes}}=\mathfrak{p}(\beta_0)$. Data constructed using the above scheme and the data used as a template may have very different characteristics. Figure 2.2 illustrates this.

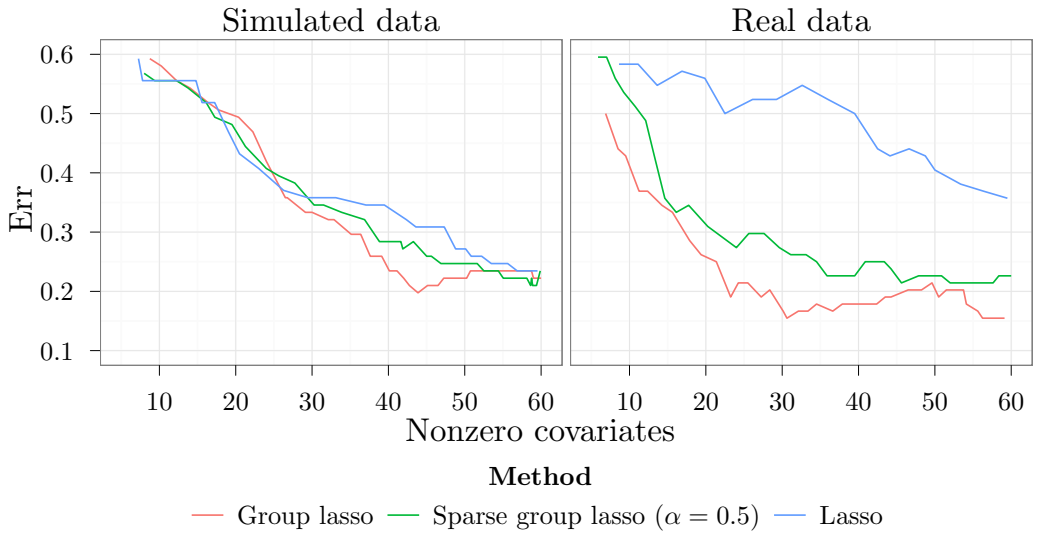


Figure 2.2: The expected generalization error as a function of the expected number of covariates included in the model for three different methods, as estimated by 8-fold cross validation. The methods were applied to a simulated data set (left) and the corresponding real data set (right). The real data set is the *Childhood Leukemia* data set. The number of samples simulated were equal to the number of samples in the real data set. The group lasso, sparse group lasso and lasso methods will be discussed in the following chapters.

2.5.1 Computing the generalization error

Using that the distribution of (X, Y) is discrete, we have that

$$\begin{aligned}\int r(x,y)\,\mathrm{d}F_{\mathrm{sim}}(x,y) &= \sum_{i=1}^{N_2}\sum_{y=1}^{K} r(x_i,y) f_{\mathrm{sim}}(x_i,y) \\ &= \frac{1}{N_2}\sum_{i=1}^{N_2}\sum_{y=1}^{K} r(x_i,y)\mathfrak{p}_y^{\mathrm{Bayes}}(x_i) \end{aligned} \tag{2.11}$$

for any function $r(x, y)$. Using formula (2.11) we may compute the generalization error of any classifier $\mathfrak{p}$. We have that

$$\begin{aligned}\mathrm{err}(\mathfrak{p}) &= \mathrm{E}\left[L(\mathfrak{p}(X), Y)\right] \\ &= \int L(\mathfrak{p}(x), y)\,\mathrm{d}F_{\mathrm{sim}}(x,y) \\ &= \frac{1}{N_2}\sum_{i=1}^{N_2}\sum_{y=1}^{K} L(\mathfrak{p}(x_i), y)\mathfrak{p}_y^{\mathrm{Bayes}}(x_i).\end{aligned}$$

And the variance of the loss is

$$\mathrm{Var}\left[L(\mathfrak{p}(X), Y)\right] = \frac{1}{N_2}\sum_{i=1}^{N_2}\sum_{y=1}^{K}(L(\mathfrak{p}(x_i), y) - \mathrm{err}(\mathfrak{p}))^2\mathfrak{p}_y^{\mathrm{Bayes}}(x_i).$$

Chapter 3

Learning

3.1 Introduction

Given a set of examples, classes and observed covariates, the question arise on how we should *learn* or *estimate* a classifier from the presented examples. A method or procedure for learning a classifier from data is called a *supervised learning method*, we will define this notion precisely. Empirical risk minimization and penalized empirical risk minimization are examples of supervised learning methods. Having defined supervised learning we will also discuss various associated statistics as for example the expected generalization error and the expected number of covariates.

In section 3.3 we briefly discuss model assessment and selection and comparison of different learning methods. And in section 3.4 we discuss error estimation using subsampling procedures as for example cross validation. We will define a notion of *M-subsampling* procedures, of which cross validation is an example, and show that these procedures give unbiased estimates of the expected generalization error. We will also briefly investigate the variance of these estimators.

3.2 Supervised learning

A supervised learning method is a collection of procedures for constructing a classifier from data $D \in (\mathbb{R}^p \times \mathcal{S}_K)^N$ – one procedure for each $N \in \mathbb{N}$. This can be modeled by a collection of functions. We therefore define:

Definition 9 (Supervised learning method)**.** *A supervised learning method $\mathcal{A}$ is a sequence of functions $\{\mathcal{A}_N\}_{n\in\mathbb{N}}$ each with signature*

$$\mathcal{A}_N : (\mathbb{R}^p \times \mathcal{S}_K)^N \times \mathbb{R}^p \to \Delta^K$$

For a data set $D \in (\mathbb{R}^p \times \mathcal{S}_K)^N$ – of size N – we will use the notation $\mathcal{A}_N(D)$ for the function $\mathcal{A}_N(D) : \mathbb{R}^p \to \Delta_K$ defined by

$$\mathcal{A}_N(D)(x) = \mathcal{A}_N(D, x).$$

Furthermore the classifier obtained by applying $\mathcal{A}$ on the data D is denote by $\mathcal{A}(D) = \mathcal{A}_N(D)$. We shall say that the classifier $\mathcal{A}(D)$ is obtained by training on D. Moreover the data D is called the *training data*.

A *parameter estimator* is an estimator for the parameters in a parametric model. This concept is captured in the following definition.

Definition 10 (Parameter estimator)**.** *A* parameter estimator *for a parametric model for classification $\mathfrak{p} : B \times \mathbb{R}^p \to \Delta^K$ is a sequence of functions $\{\hat{\beta}_N\}_{n\in\mathbb{N}}$ each with the signature*

$$\hat{\beta}_N : (\mathbb{R}^p \times \mathcal{S}_K)^N \to B.$$

If a parametric model for classification is given then we may construct a supervised learning method by composing the parametric model with a parameter estimator. This composition is as shown in the following diagram:

$$\begin{array}{ccc} (\mathbb{R}^p \times \mathcal{S}_K)^N \times \mathbb{R}^p & \xrightarrow{\hat{\beta}_N \times \mathrm{id}} & B \times \mathbb{R}^p \\ & \searrow^{\mathcal{A}_N} & \downarrow \mathfrak{p} \\ & & \Delta^K \end{array}$$

Example 3 (Empirical risk minimization)**.** *Empirical risk minimization is an example of a supervised learning method. Given a parametric model $\mathfrak{p}$, a loss function and data $D \in (\mathbb{R}^p \times \mathcal{S}_K)^N$ we may construct the empirical risk $\hat{R}_D(\beta)$. Empirical risk minimization Vapnik [19] then estimates a model by*

$$\hat{\beta} = \arg\min_{\beta\in B} \hat{R}_D(\beta). \tag{3.1}$$

And a supervised learning method $\mathcal{A}_N$ may be constructed by setting

$$\mathcal{A}_N(D)(x) = \mathfrak{p}(\hat{\beta})(x).$$

3.2.1 Model characteristics

We are often interested in comparing learning methods. We may, for example, be interested in comparing the expected generalization errors of different methods when applied to data drawn from some specified population. Moreover other characteristics of the estimated parameters and/or classifier may be of interest, as for example the number of covariates used in the model. We therefore need a notion of *model characteristics*, that broadens the notion of generalization error – model characteristic will be defined below.

For any supervised learning method the expected error of the resulting classifier is called the expected generalization error.

Definition 11 (Expected generalization error)**.** *The* expected generalization error *of a supervised learning method $\mathcal{A}$ is*

$$\mathrm{Err}(N) \stackrel{def}{=} \mathrm{E}(L(\mathcal{A}_N(\mathfrak{D})(X), Y)).$$

Note that the expected generalization error is a function of the number of samples in $\mathfrak{D}$. We will however often use the notation Err, instead of $\mathrm{Err}(N)$, when the number of samples is implicit. The expected generalization error is the expectation of the true error taken over training data, as is seen from the relation

$$\mathrm{Err} = \mathrm{E}\left[\mathrm{E}(L(\mathcal{A}_N(\mathfrak{D})(X), Y) \mid \mathfrak{D})\right] = \mathrm{E}(\mathrm{err}[\mathcal{A}(\mathfrak{D})]).$$

The generalization error is an example of a model characteristic, we may however consider various other characteristics of a parametric model $\beta \in B$. We wish to broaden the concept of expected generalization error so that we can speak of the expected characteristic of a model. A model characteristic is induced by a function T, and we say that:

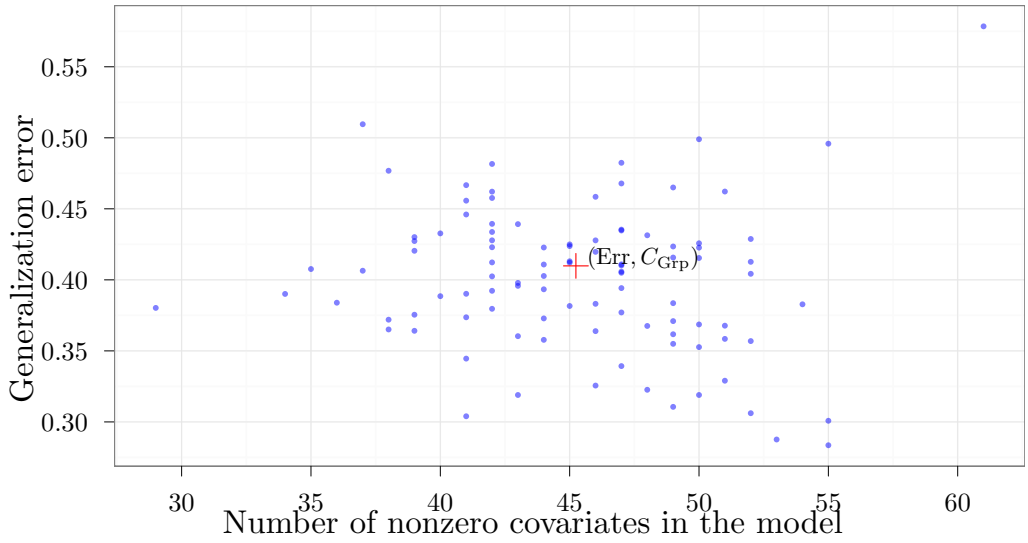


Figure 3.1: The group lasso estimator (with fixed λ) applied to simulated data sets of size 40, all data sets were simulated using the same distribution. This distribution were of the from (2.10) and the *Childhood Leukemia* data set were used as a template. With the model of the distribution β_0 obtained using the group lasso estimator (see Example 5). Each of the 100 dots represent the characteristics of a classifier obtained by training on one of the simulated data sets. The cross show the expected generalization error and expected number of covariates in the model, as obtained by the mean of 1k classifiers.

Definition 12. *For a function $T : B \times (\mathbb{R}^p \times \mathcal{S}_K) \to \mathbb{R}$ the by T induced model characteristic is the function $\chi_T : B \to \mathbb{R}$ defined by*

$$\chi_T(\beta) \stackrel{def}{=} \mathrm{E}\left[T(\beta, X, Y)\right].$$

Note that model characteristic concept is only defined for parametric models, contrary to the expected generalization error which is defined for all supervised learning methods. Examples of model characteristics are the true error, the number of covariates used in the model and the number of nonzero parameters in the model.

The expectation of a model characteristic over training data is called the *expected model characteristic*, this concept broadens the concept of expected generalization error. Figure 3.1 illustrates the concept of expected model characteristics. The definition is:

Definition 13. *The expected model characteristic for an estimator $\hat{\beta}$ is*

$$C_T \stackrel{def}{=} \mathrm{E}\left(\chi_T(\hat{\beta})\right) = \mathrm{E}\left(\chi_T\left[\hat{\beta}(\mathfrak{D})\right]\right).$$

For a loss L we may let $T_L(\beta, x, y) = L(\mathfrak{p}(\beta)(x), y)$ then the induced model characteristic is the error, that is

$$\chi_{T_L}(\beta) = \mathrm{err}(\beta).$$

Furthermore the expected model characteristic induced by T is the generalization error, hence

$$C_{T_L} = \mathrm{E}(\mathrm{err}[\mathcal{A}(\mathfrak{D})]) = \mathrm{Err}\,.$$

If B is a vector space then another example of a model characteristic is the number of nonzero parameters. This characteristic is induced by the function

$$\mathrm{Par}(\beta) \stackrel{\mathrm{def}}{=} \sum_{i=1}^{Kp} 1(\beta_i \neq 0).$$

Evidently Par is independent of X and Y – it only depends on the model β. It follows that the model characteristic $\chi_{\mathrm{Par}} = \mathrm{Par}(\beta)$. The expected number of nonzero parameters is

$$C_{\mathrm{Par}} = \mathrm{E}(\mathrm{Par}(\hat{\beta})) = \sum_{i=1}^{Kp} P(\hat{\beta}_i \neq 0).$$

The vector space B may posses some additional structure. We are primarily interested in the case where the parameters may be naturally grouped. That is there is a natural decomposition of the parameter space $B = \mathbb{R}^n$

$$\mathbb{R}^n = \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_m}$$

into $m \in \mathbb{N}$ groups. The groups having dimensions $n_i \in \mathbb{N}$ for $i = 1, \ldots, m$, hence $n = n_1 + \cdots + n_m$. For a vector $\beta \in \mathbb{R}^n$ we write $\beta = (\beta^{(1)}, \ldots, \beta^{(m)})$ where $\beta^{(1)} \in \mathbb{R}^{n_1}, \ldots, \beta^{(m)} \in \mathbb{R}^{n_m}$. For $J = 1, \ldots, m$ we call $\beta^{(J)}$ the J'th *group* of β. We use the notation $\beta_i^{(J)}$ to denote the i'th coordinate of the J'th group of β, whereas β_i is the i'th coordinate of β.

When a grouping of the parameters is present it is natural to consider the number of nonzero groups, i.e.

$$\mathrm{Grp}(\beta) \stackrel{\mathrm{def}}{=} \sum_{I=1}^{m} 1(\beta^{(I)} \neq 0).$$

The number of nonzero groups is a model characteristic, induced by itself, hence $\chi_{\text{Grp}} = \text{Grp}(\beta)$. The expected number of nonzero groups is

$$C_{\text{Grp}} = \text{E}(\text{Grp}(\hat{\beta})) = \sum_{J=1}^{m} P(\hat{\beta}^{(J)} \neq 0).$$

For linear models the parameter space B is structured as $K \times p$ matrices, with each column corresponding to a covariate. That is K parameters per covariate, we may group these K parameters together – we will consider this the standard grouping for linear models. Using this grouping a parameter $\beta \in B$ can be written as a block matrix in the following way:

$$\beta = \left(\beta^{(1)} \cdots \beta^{(p)}\right).$$

For linear models, with the standard grouping of the parameters, the number of covariate used in the model $\beta \in B$ is the number of nonzero groups $\text{Grp}(\beta)$.

Remark 1. *The above definition of expected model characteristic can, due to limited number of samples, be somewhat unhandy for practical use. When estimating the expected model characteristic (or expected generalization error) then in practice we may need to condition on the number of samples in each class. Given a random data set $\mathfrak{D}$ of size N we may define the K random variables*

$$N_k \overset{def}{=} \sum_{i=1}^{N} 1(Y_i = k) \quad \textit{for } k \in \mathcal{S}_K.$$

That is N_k is the number of samples of class k. For a model characteristic χ_T and $a = (a_1, \ldots, a_K) \in \mathbb{N}^K$ with $a_1 + \cdots + a_K = N$ the conditional expected model characteristic *is*

$$C_{T|a} = \text{E}(\chi_T(\hat{\beta}) \mid N_1 = a_1, \ldots, N_K = a_K)$$

For most classification problems it is natural to take $a_1 = \cdots = a_K$.

3.2.2 Parametrized learners.

In many situations one supervised learning method is not enough, we may wish to try out a range of methods and then select the method that induces classifiers best suited for our needs. This is the case when we do *feature selection*, when we select the amount of regularization for *penalized empirical risk minimization* or when we adjust some parameter of the method. In order to formalize and make this precise we will define the concept of parametrized supervised learning methods.

Definition 14. *A parametrized supervised learning method is a sequence of functions $\{\mathcal{A}_N\}_{n\in\mathbb{N}}$ each with the signature*

$$\mathcal{A}_N : \Lambda \times (\mathbb{R}^p \times \mathcal{S}_K)^n \times \mathbb{R}^p \to \Delta^K$$

and such that $\mathcal{A}(\lambda)$ (i.e. $\{\mathcal{A}_N(\lambda)\}_{n\in\mathbb{N}}$) is a supervised learning method for each $\lambda \in \Lambda$.

Like with supervised learning methods we often consider a parametrized supervised learning method as a two step procedure, first parameter estimation then application of the parametric model.

Definition 15. *A* parametrized model estimator *for a parametric model* $\mathfrak{p}$ *is a sequence* $\{\hat{\beta}_N\}_{n\in\mathbb{N}}$ *of functions each with signature*

$$\hat{\beta}_N : \Lambda \times (\mathbb{R}^p \times \mathcal{S}_K)^n \to B.$$

For a parametrized supervised learning method the expected model characteristic is a function of the λ parameter. An example of a parametrized supervised learning method is penalized empirical risk minimization;

3.2.3 Penalized empirical risk minimization

Let $\hat{R}_D(\beta)$ denote the empirical risk associated with the parametric model $\mathfrak{p}$ and data $D \in (\mathbb{R}^p \times \mathcal{S}_K)^N$. Penalized empirical risk minimization then estimates a model by

$$\hat{\beta}(\lambda) = \arg\min_{\beta\in B} \hat{R}_D(\beta) + \lambda\Phi(\beta) \tag{3.2}$$

with $\lambda > 0$ and where $\Phi : B \to \mathbb{R}_+$ is the regularization term or penalty. The scalar λ is sometimes called the *amount of regularization*. A parametrized supervised learning method $\mathcal{A}$ is defined by setting

$$\mathcal{A}_N(\lambda)(D)(x) = \mathfrak{p}(\hat{\beta}(\lambda))(x)$$

for $D \in (\mathbb{R}^p \times \mathcal{S}_K)^N$.

For penalized empirical risk minimization the choice of penalty is essential, different penalties will result in methods with different characteristics. This is evident by looking at Figure 3.3 and 3.4.

Two important examples of penalties are:

Example 4 (Lasso)**.** *Lasso (Tibshirani [15]) for linear classification estimates models by penalized empirical risk minimization with the penalty*

$$\Phi(\beta) = \sum_{i=1}^{K}\sum_{j=1}^{p} \xi_{ij}\,|\beta_{ij}|$$

with weights $\xi_{ij} \in [0,\infty)$*. In the case that* $\xi_{ij} = 1$ *for all* $i = 1,\ldots,K$ *and all* $j = 1,\ldots,p$ *the penalty is the 1-norm of the vector* $\mathrm{vec}(\beta)$*. The lasso penalty has been considered for classification for some time, see for example Zhu and Hastie [24].*

Example 5 (Group lasso)**.** *The standard group lasso for linear classification models (Vincent and Hansen [21]) estimates models by regularized risk minimization with the penalty*

$$\Phi(\beta) = \sum_{J=1}^{p} \gamma_J \left\|\beta^{(J)}\right\|_2$$

with weights $\gamma_J \in [0,\infty)$*. Do not confuse the group lasso with ridge regression, the latter being* $\Phi(\beta) = \|\mathrm{vec}(\beta)\|_2^2$*, that is the squared 2-norm. The selection effect of the group lasso comes from the non differentiability at zero of the 2-norm, which is removed by squaring. Group lasso penalties has also been used to group covariates in combination with logistic regression (Meier et al. [11]), this is however a different use of group lasso than the one presented in this example.*

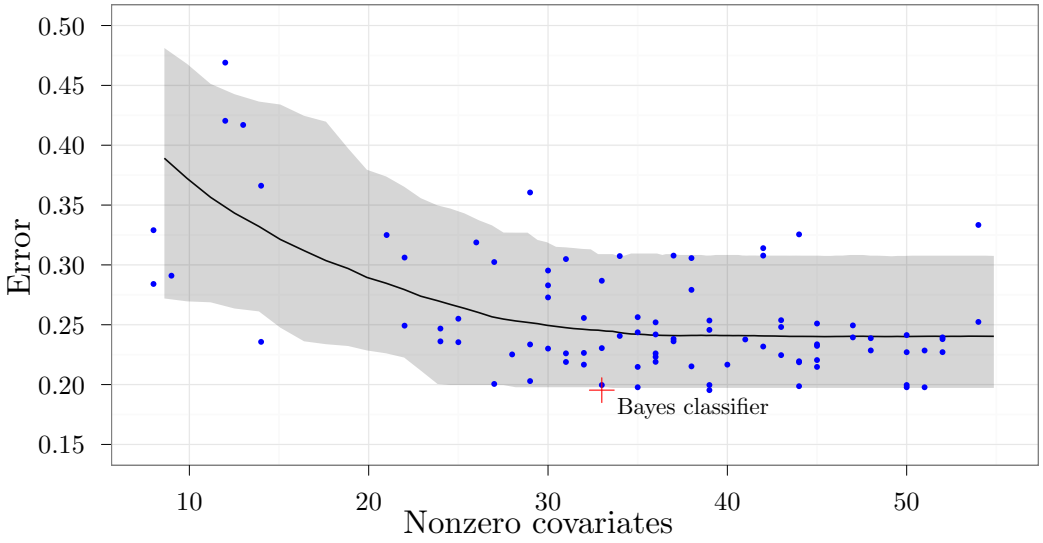


Figure 3.2: The group lasso estimator applied to simulated data sets of size 80, all data sets were drawn from the same simulated population. The distribution, used for the simulation, is of the from (2.10). The *Brain Tumor* data set were used as template, using the group lasso estimator to obtain the model β_0 used for simulation. For each λ in a pre-computed sequence $\{\lambda_1, \ldots, \lambda_{100}\}$ of values 1k classifiers were trained on individually simulated data sets. For each λ one classifier was randomly selected and marked by a dot with coordinates $(\mathrm{Grp}(\beta), \mathrm{err}(\beta))$, where β denotes the model of the classifier. The curve is $(C_{\mathrm{Grp}}(\lambda), \mathrm{Err}(\lambda))$, as estimated from all 1k classifiers. The shaded area marks the 5% and 95% empirical quantiles of the distributions $\mathrm{err}(\hat{\beta}(\lambda))$ for $\lambda \in \{\lambda_1, \ldots, \lambda_{100}\}$. The cross marks the characteristics of the Bayes classifier, that is the model β_0.

3.3 Model assessment and selection

In this section we will briefly discuss three model assessment and selection tasks. We assume that the distribution of the data is specified. The tasks are:

1. *Method comparison* – comparison of the characteristics of two or more methods.
2. *Model selection* – selection of an model with optimal performance.
3. *Model assessment* – assessment of the characteristics of a fixed model.

Model selection and assessment may seem to be similar, the difference is that for model selection the only concern is selection of an *optimal* model, whereas for model assessment a concrete estimate of, for example, performance is sought. To be precise we should say *model selection for estimation* to distinct it from *model selection for identification* (Arlot and Celisse [1]). Model selection for identification aims at selecting the *true model* whereas model selection for estimation aims at selecting an optimal model in terms of some measure of error.

3.3.1 Method comparison

For method comparison we are interested in comparing the distributions of the model characteristics $\chi_T(\mathcal{A}(\mathfrak{D}))$ and $\chi_T(\mathcal{B}(\mathfrak{D}))$ with $\mathfrak{D}$ is a random data set. Although the entire distributions many be of interest, we usually focus on one-number summaries of the distributions. One-number summaries are often easier to interpret and estimate. For example, we may compare expected model characteristics.

For parametric methods expected model characteristics are functions of the parameter λ that parameterize the methods. Considering two parametric methods $\mathcal{A}$ and $\mathcal{B}$, the question arise on how we should compare the two curves

$$c_A : \lambda \to C_T(\mathcal{A}(\lambda)) \quad \text{and} \quad c_B : \lambda \to C_T(\mathcal{B}(\lambda)).$$

We cannot simply compare the curves as there, for given λ, may not be any natural relation between $c_A(\lambda)$ and $c_B(\lambda)$. We could compare the maximum and/or minimum of the curves. This would seem appropriate if we seek the methods producing the most extreme models with respect to, for example, the error rate. However, in many situations we not only seek a model with a low error rate, but one which also has a low model complexity.

Consider the case where we have two model characteristics T_1 and T_2. Typically T_1 is a measure of model complexity and T_2 a measure of error (i.e. a loss function). We are then interested in comparing the distributions

$$\left(\chi_{T_1}(\hat{\beta}_{\mathcal{A}}(\lambda_A)), \chi_{T_2}(\hat{\beta}_{\mathcal{A}}(\lambda_A))\right) \text{ and } \left(\chi_{T_1}(\hat{\beta}_{\mathcal{B}}(\lambda_B)), \chi_{T_2}(\hat{\beta}_{\mathcal{B}}(\lambda_B))\right).$$

Note that the distributions are parametrized with parameter $\lambda_A \in \Lambda_A$ and $\lambda_B \in \Lambda_B$ respectively.

Take T_1 to be the number of covariates included in the model and T_2 the 01 loss. We may then compare the methods by comparing their corresponding *characteristic curves*, that is by comparing the parametrized curves

$$\left(\mathrm{Grp}(\hat{\beta}_A(\lambda_{\mathcal{A}})), \mathrm{Err}(\hat{\beta}_A(\lambda_{\mathcal{A}}))\right) \quad \text{and} \quad \left(\mathrm{Grp}(\hat{\beta}_{\mathcal{B}}(\lambda_{\mathcal{B}})), \mathrm{Err}(\hat{\beta}_{\mathcal{B}}(\lambda_{\mathcal{B}}))\right)$$

with $\lambda_{\mathcal{A}}$ and $\lambda_{\mathcal{B}}$ in respectively $\Lambda_{\mathcal{A}}$ and $\Lambda_{\mathcal{B}}$. See Figure 3.2 for an example of a characteristic curve.

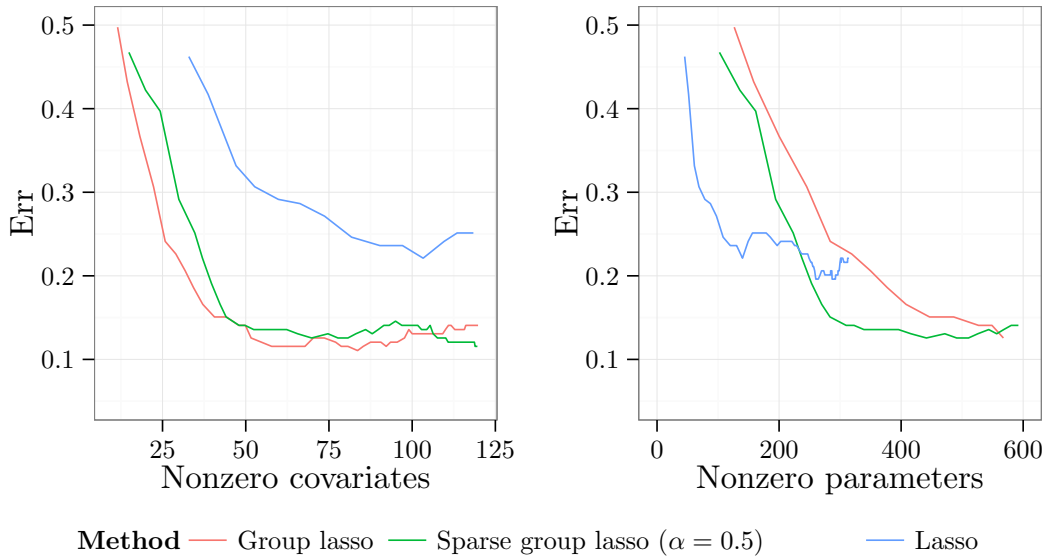


Figure 3.3: Characteristic curves for the group lasso (see Example 5), the lasso (see Example 4) and the sparse group lasso with $\alpha = 0.5$ (see section 4.6) applied to the *Primary Cancers* data set. For all three methods the shown characteristic curves are $(C_{\text{Grp}}, \text{Err})$ (left) and $(C_{\text{Par}}, \text{Err})$ (right). The curves were estimated by 10-fold cross validation on a λ-sequence of length 100.

In Figure 3.2 we used simulated data and could therefore compute the exact characteristic curve, in practice however, we will have to estimate the characteristic curves – that is to estimate the expected model characteristics. This can often be done using either a cross validation or a subsampling scheme, as discussed in section 3.4.2. In Figure 3.3 we estimated, using cross validation, characteristic curves for three different learning methods applied to the *Primary Cancers* data set. Looking at Figure 3.3 it is apparent that the three learning methods has different characteristics.

Learning curves

The expected generalization error is a function of the number of samples N in the training data. The *learning curve* is the graph

$$(N, \text{Err}(N)) .$$

In the case of a parametrized learning method we could consider the curve

$$\left(N, \min_{\lambda \in \Lambda} \text{Err}(\hat{\beta}_N(\lambda))\right) .$$

This curve shows how the optimal model depends on the number of training samples. Another interesting curve is, for example, the curve

$$\left(C_{\text{Grp}}(\hat{\beta}_N(\lambda_N^*)), \text{Err}(\hat{\beta}_N(\lambda_N^*))\right)$$

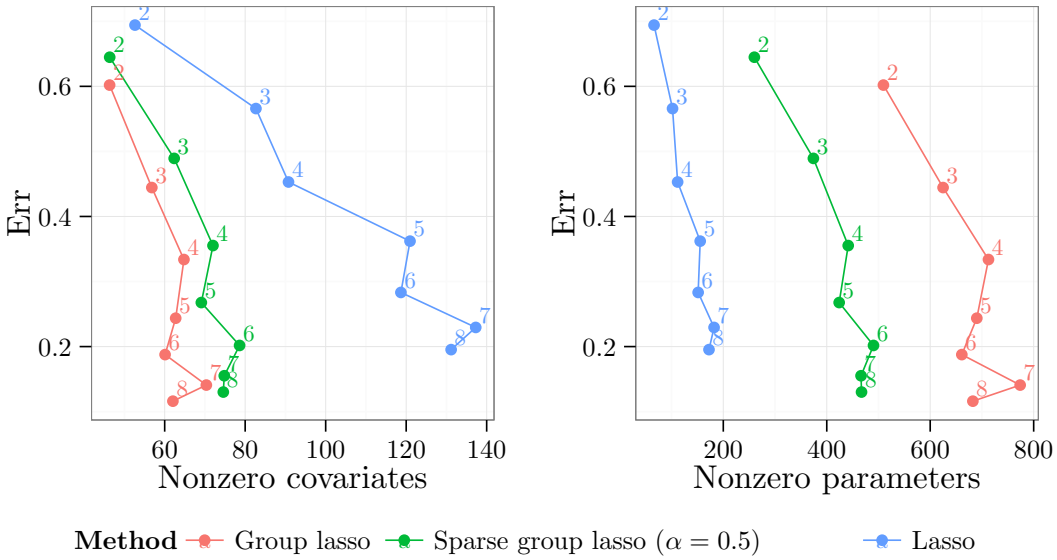


Figure 3.4: Learning curves for the group lasso (see Example 5), the lasso (see Example 4) and the sparse group lasso with $\alpha = 0.5$ (see section 4.6) applied to the *Primary Cancers* data set. The curves shown are $(C_{\text{Grp}}(\hat{\beta}_N(\lambda_N^*)), \text{Err}(\hat{\beta}_N(\lambda_N^*))$ (left) and $(C_{\text{Par}}(\hat{\beta}_N(\lambda_N^*)), \text{Err}(\hat{\beta}_N(\lambda_N^*))$ (right), the number of samples per class is shown as numbers on the plots. The lambda parameter λ_N^* is chosen such that the estimator $\hat{\beta}_N(\lambda_N^*)$ has lowest expected generalization error among the estimators $\{\hat{\beta}_N(\lambda_N^*)\}_{\lambda \in \Lambda}$. The characteristics were estimated by subsampling (see section 3.4) with 200 subsamples. For each subsample a training set of size N and a test set with 8 samples per class was randomly chosen without overlap.

parametrized by N and where $\lambda_N^* \stackrel{\text{def}}{=} \arg\min_{\lambda \in \Lambda} \text{Err}(\hat{\beta}_N(\lambda))$. Examples of such learning curves, for three different methods, can be seen in Figure 3.4. Differences between the three methods are apparent.

3.3.2 Model selection

Model selection is a large research subject and we will here only briefly discuss it, for a more throughout discussion see for example Arlot and Celisse [1] or Hastie et al. [7]. The main question of model selection is: given data $D \in (\mathbb{R}^p \times \mathcal{S}_K)^N$ which λ should we choose in order to achieve the optimal error of $\mathcal{A}(\lambda)(D)$, i.e. in order to minimize $\text{err}(\mathcal{A}(\lambda)(D))$.

Consider now the *error function*

$$\lambda \to \text{err}\left[\mathcal{A}(\lambda)(D)\right]. \tag{3.3}$$

It is interesting that when estimating the parameters of a multinomial model using sparse group lasso the error function (3.3) for the log-likelihood loss seems always to be quasiconvex – on $\mathbb{R}$ the quasiconvex functions are exactly the monotone and unimodal functions. In fact the error function seems mostly strictly convex near the minimizer.

Consider the log-likelihood loss and assume, for the following discussion, that the error function (3.3) is strictly convex. Then there exist a unique minimizer λ^* of the error function.

On approach to model selection is to try and estimate λ^*. If we can estimate λ^* well then we may obtain a model with near optimal error with respect to the likelihood loss.

However, we are often interested in the misclassification error, i.e. the 01 loss. The 01 loss error function is not quasiconvex, in fact it may oscillate and it is in general hard to estimate a minimizer. Unfortunately the minimizer of the likelihood loss error function may not be near a minimizer of the 01 loss error function. Moreover we don't know the error function.

It is in general difficult to estimate the true error and therefore also the error function. We can, however, estimate the expected generalization error fairly well. We could therefore attempt to estimate a minimizer λ^* of the error function by

$$\hat{\lambda}^* = \arg\min_{\lambda} \widehat{\mathrm{Err}}(\lambda).$$

where $\widehat{\mathrm{Err}}$ is an estimate of the expected generalization error.

Given an parametrized supervised learning method $\mathcal{A}$ and a model selection method, that is an estimator $\hat{\lambda}^*$, we may construct a supervised learning method by

$$\mathcal{A}^*(D) \stackrel{\mathrm{def}}{=} \mathcal{A}(\hat{\lambda}^*)(D).$$

The characteristics of the combined method $\mathcal{A}^*$ will in general have to be estimated and will depend on both the characteristics of the method $\mathcal{A}$ and the model selection method.

3.3.3 Assessment of model characteristics

Given data $D \in (\mathbb{R}^p \times \mathcal{S}_K)^n$ we are often interested in various model characteristics of an estimated model $\hat{\beta}(D)$. Some characteristics may be directly accessibly, for example the number of nonzero parameters or the number of nonzero groups. However other characteristics like the true error needs to be estimated.

The true error can be estimated using an independent test set. However if no, or only a small, independent test set is available it is in practice impossible to obtain a unbiased estimate of the true error. We may instead – by cross validation or other subsampling procedures – obtain a unbiased estimate of the expected generalization error. The problem, with this approach, is that we do not know the variance of $\mathrm{err}[\mathcal{A}(\mathfrak{D})]$ nor the variance of our estimation procedure. Hence the estimated expected generalization error may in worst case be far away from the true error. Although, in practice, it is often found that the cross validation estimate agrees well with estimates obtained using an independent test set.

3.4 Error estimation

In this section we consider ways of estimating the model characteristic $\chi_T(\hat{\beta})$ and the expected model characteristic $C_T(\hat{\beta})$ of an estimator $\hat{\beta}$. An unbiased estimate of the model characteristic can be obtained by an independent test as described below. For the expected model characteristic various subsampling procedures may be used. Given an estimator $\hat{\beta}$ we define:

Definition 16 (Sample model characteristic)**.** *For data sets D_{train} and D_{test} of size respectively N_{test} and N_{train} the sample model characteristic is*

$$\hat{\chi}_T(D_{\mathrm{train}}, D_{\mathrm{test}}) \stackrel{def}{=} \frac{1}{N_{\mathrm{test}}} \sum_{i=1}^{N_{\mathrm{test}}} T(\hat{\beta}(D_{\mathrm{train}}), X_i, Y_i).$$

Note that the sample model characteristic is a function of two data sets: a training and a test data set. Moreover the estimator $\hat{\beta}$ is implicit in the definition.

3.4.1 Estimation by independent test

For a training data set D and a random test data set $\mathfrak{D}_{\text{test}}$ the the random variable

$$\hat{\chi}_T(D, \mathfrak{D}_{\text{test}}).$$

is called the *test characteristic* of the estimator $\hat{\beta}(D)$. By the central limit theorem

$$\frac{(\hat{\chi}_T(D, \mathfrak{D}_{\text{test}}) - \mu)}{\sigma/\sqrt{N_{\text{test}}}} \rightsquigarrow N(0,1) \text{ as } N_{\text{test}} \to \infty$$

where $\mu = \chi_T(\hat{\beta})$ and $\sigma^2 = \text{Var}(T(\hat{\beta}, X, Y))$. That is the test characteristic is a unbiased estimator of the model characteristic and the estimated standard error of the test characteristic is

$$\hat{\text{se}} = \sqrt{\frac{\hat{\sigma}^2}{N_{\text{test}}}}.$$

This implies that if N_{test} is sufficiently high we may assume that

$$\frac{\hat{\chi}_T(D, \mathfrak{D}_{\text{test}}) - \mu}{\hat{\text{se}}} \approx N(0,1).$$

So approximate 95% confidence interval for the model characteristic is therefore

$$\hat{\chi}_T(D, \mathfrak{D}_{\text{test}}) \pm 1.96\hat{\text{se}}.$$

If the training and test data are dependent then the estimated model characteristic may be severely biased, this is in particular the case for high dimensional classification. The *training characteristic* of $\hat{\beta}$ is

$$\hat{\chi}_T(D, D).$$

For the most part we are interested in estimating the error, i.e. T is a loss, of the classifier model. In this case the test characteristic is simply called the *test error* and the training characteristic the *training error*. In Figure 3.5 we see that the training error is highly over optimistic, the misclassification error reaches approximately 0 when more than 500 features are included in the model. We also see that the optimism (the gab between the training and test error) increases with the number of covariates include in the model.

3.4.2 Estimation by subsampling procedures

In this section we consider subsampling procedures for estimating the expected model characteristic.

Consider the random data set

$$\mathfrak{D} = ((X_1, Y_1), \ldots, (X_N, Y_N)).$$

of size N. Let S_N denote the set of all permutations of N elements, then for a permutation $\tau \in S_N$ define the two data sets

$$\mathfrak{D}^{\tau}_{\text{train}} = \left((X_{\tau(1)}, Y_{\tau(2)}), \ldots, (X_{\tau(N_{\text{train}})}, Y_{\tau(N_{\text{train}})})\right)$$

and

$$\mathfrak{D}^{\tau}_{\text{test}} = \left((X_{\tau(N_{\text{train}}+1)}, Y_{\tau(N_{\text{train}}+1)}), \ldots, (X_{\tau(N)}, Y_{\tau(N)})\right)$$

of size respectively N_{train} and $N_{\text{test}} = N - N_{\text{train}}$. We are now ready to define a general notation of subsampling procedures, the idea is that a subsampling procedure is specified by a distribution on the random permutations. We first define the notion of M-subsample:

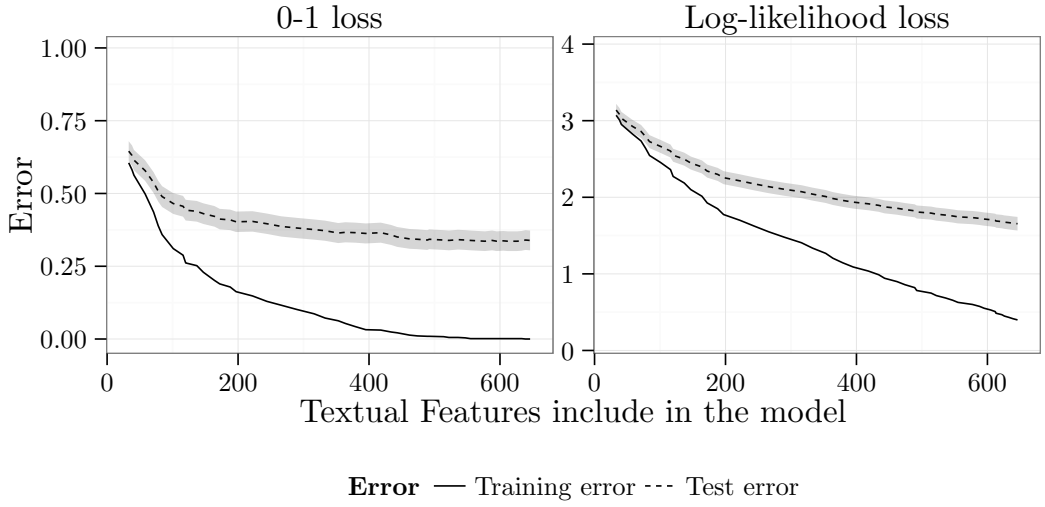


Figure 3.5: The test and training error of the *Amazon Reviews* data set, the models were estimated using group lasso. The error measured with the 01 loss (left) and the log-likelihood loss (right) is shown. The data set were split in two half, keeping class ratios approximately fixed, one half were used as a training set and the other as a test set. The shaded area indicates the approximate normal-based 95% confidence interval.

Definition 17 (M-subsample)**.** *A M-subsample of $\mathfrak{D}$ is the collection*

$$\mathfrak{D}^{\tau_1}_{train}, \mathfrak{D}^{\tau_1}_{test}, \ldots, \mathfrak{D}^{\tau_M}_{train}, \mathfrak{D}^{\tau_M}_{test}$$

of $2M$ data sets where $\tau_1, \ldots, \tau_M \in S_N$ are random permutations.

We will not assume independence of the random permutations $\tau_1, \ldots, \tau_M$. We will however assume that the sequence $\tau_1, \ldots, \tau_M$ is *exchangeable*, i.e. that

$$P(\tau_1, \ldots, \tau_M) = P(\tau_{\gamma(1)}, \ldots, \tau_{\gamma(M)})$$

for any permutation $\gamma \in S_M$. A subsampling procedure is then specified by choosing an exchangeable distribution for the random permutations $(\tau_1, \ldots, \tau_M)$. Cross validation and standard subsampling are exchangeable procedures, i.e. they are defined by an exchangeable sequence of random permutations $\tau_1, \ldots, \tau_M$. Cross validation and standard subsampling will be discussed below.

Let $\hat{C}_T^{\tau_i} \stackrel{\text{def}}{=} \hat{\chi}_T(\mathfrak{D}^{\tau_i}_{\text{train}}, \mathfrak{D}^{\tau_i}_{\text{test}})$ and define the sample expected model characteristic by

$$\hat{C}_T \stackrel{\text{def}}{=} \frac{1}{M} \sum_{i=1}^{M} \hat{C}_T^{\tau_i}.$$

Note that $\hat{C}_T$ depends on the joint distribution of the permutations $(\tau_1, \ldots, \tau_M)$, hence the distribution of $\hat{C}_T$ will differ depending on the subsampling procedure we choose. However, as the following proposition show, $\hat{C}_T$ is always a unbiased estimator of the expected model characteristic C_T.

Proposition 2. *It holds that* $\mathrm{E}(\hat{C}_T) = C_T$.

Proof. For a permutation $\tau \in S_N$ we have

$$\begin{aligned}
\mathrm{E}(\hat{\chi}_T(\mathfrak{D}^{\tau}_{\text{train}}, \mathfrak{D}^{\tau}_{\text{test}})) &= \mathrm{E}[T(\hat{\beta}(\mathfrak{D}^{\tau}_{\text{train}}), X_{\tau(N_{\text{train}}+1)}, Y_{\tau(N_{\text{train}}+1)})] \\
&= \mathrm{E}\,\mathrm{E}[T(\hat{\beta}(\mathfrak{D}^{\tau}_{\text{train}}), X_{\tau(N_{\text{train}}+1)}, Y_{\tau(N_{\text{train}}+1)}) \mid \mathfrak{D}^{\tau}_{\text{train}})] \\
&= \mathrm{E}(\chi_T(\mathfrak{D}^{\tau}_{\text{train}}))) \\
&= C_T
\end{aligned}$$

This implies that for any $i \in \{1, \ldots, M\}$

$$\mathrm{E}\,\hat{\chi}_T(\mathfrak{D}^{\tau_i}_{\text{train}}, \mathfrak{D}^{\tau_i}_{\text{test}}) = C_T$$

hence $\mathrm{E}(\hat{C}_T) = C_T$. □

The following proposition gives some insight into the components of the variance of the distribution of the estimator $\hat{C}_T$.

Proposition 3. *If the sequence of random permutations $\tau_1, \ldots, \tau_M$ is exchangeable then*

$$\mathrm{Var}(\hat{C}_T) = \frac{1}{N_{\text{test}} M} \mathrm{E}(\sigma^2) + \frac{1}{M} \mathrm{Var}(\mu) + \left(1 - \frac{1}{M}\right) \mathrm{Cov}(\hat{C}_T^{\tau_1}, \hat{C}_T^{\tau_2}).$$

where $\mu = \chi_T(\hat{\beta})$ and $\sigma^2 = \mathrm{Var}(T(\hat{\beta}, X, Y) \mid \mathfrak{D}_{\text{train}})$.

Proof. First note that since the data set $\mathfrak{D}$ is i.i.d. it follows that

$$\begin{aligned}\operatorname{Var}(\hat{C}_T^{\tau_i} \mid \tau_i, \mathfrak{D}_{\text{train}}^{\tau_i}) &= \operatorname{Var}(\hat{\chi}_T(\mathfrak{D}_{\text{train}}^{\tau_i}, \mathfrak{D}_{\text{test}}^{\tau_i}) \mid \tau_i, \mathfrak{D}_{\text{train}}^{\tau_i}) \\ &= \frac{1}{N_{\text{test}}} \operatorname{Var}(T(\hat{\beta}(\mathfrak{D}_{\text{train}}), X, Y) \mid \mathfrak{D}_{\text{train}}) \\ &= \frac{1}{N_{\text{test}}} \sigma^2.\end{aligned}$$

In particular $\operatorname{Var}(\hat{C}_T^{\tau_i} \mid \tau_i, \mathfrak{D}_{\text{train}}^{\tau_i})$ is independent of τ_i. Second

$$\begin{aligned}\operatorname{E}(\hat{C}_T^{\tau_i} \mid \tau_i, \mathfrak{D}_{\text{train}}^{\tau_i}) &= \operatorname{E}(\hat{\chi}_T(\mathfrak{D}_{\text{train}}^{\tau_i}, \mathfrak{D}_{\text{test}}^{\tau_i}) \mid \tau_i, \mathfrak{D}_{\text{train}}^{\tau_i}) \\ &= \operatorname{E}\left(T(\hat{\beta}(\mathfrak{D}_{\text{train}}), X, Y) \mid \mathfrak{D}_{\text{train}}\right) \\ &= \mu.\end{aligned}$$

It follows that

$$\begin{aligned}\operatorname{Var}(\hat{C}_T^{\tau_i}) &= \operatorname{E}\left[\operatorname{Var}(\hat{C}_T^{\tau_i} \mid \tau_i, \mathfrak{D}_{\text{train}}^{\tau_i})\right] + \operatorname{Var}\left[\operatorname{E}(\hat{C}_T^{\tau_i} \mid \tau_i, \mathfrak{D}_{\text{train}}^{\tau_i})\right] \\ &= \frac{1}{N_{\text{test}}} \operatorname{E}(\sigma^2) + \operatorname{Var}(\mu).\end{aligned}$$

This implies

$$\begin{aligned}\operatorname{Var}(\hat{C}_T \mid \tau_1, \ldots, \tau_M) &= \frac{1}{M^2}\left[\sum_{i=1}^{M} \operatorname{Var}(\hat{C}_T^{\tau_i}) + \sum_{i \neq j} \operatorname{Cov}(\hat{C}_T^{\tau_i}, \hat{C}_T^{\tau_j} \mid \tau_1, \ldots, \tau_M)\right] \\ &= \frac{1}{N_{\text{test}} M} \operatorname{E}(\sigma^2) + \frac{1}{M} \operatorname{Var}(\mu) + \frac{1}{M^2} \sum_{i \neq j} \operatorname{Cov}(\hat{C}_T^{\tau_i}, \hat{C}_T^{\tau_j} \mid \tau_1, \ldots, \tau_M).\end{aligned}$$

Furthermore by permutational invariance of the subsampling procedure

$$\operatorname{E}\operatorname{Cov}(\hat{C}_T^{\tau_i}, \hat{C}_T^{\tau_j} \mid \tau_1, \ldots, \tau_M) = \operatorname{E}\operatorname{Cov}(\hat{C}_T^{\tau_1}, \hat{C}_T^{\tau_2} \mid \tau_1, \tau_2).$$

So since $\operatorname{E}(C_T \mid \tau_1, \ldots, \tau_M) = C_T$ is independent of $\tau_1, \ldots, \tau_M$ is follows that

$$\begin{aligned}\operatorname{Var}(C_T) &= \operatorname{E}\operatorname{Var}(C_T \mid \tau_1, \ldots, \tau_M) + \operatorname{Var}\operatorname{E}(C_T \mid \tau_1, \ldots, \tau_M) \\ &= \frac{1}{N_{\text{test}} M} \operatorname{E}(\sigma^2) + \frac{1}{M} \operatorname{Var}(\mu) + \left(1 - \frac{1}{M}\right) \operatorname{E}\operatorname{Cov}(\hat{C}_T^{\tau_1}, \hat{C}_T^{\tau_2} \mid \tau_1, \tau_2).\end{aligned}$$

Finally the statement follows by noting that by the law of total covariance

$$\operatorname{E}\operatorname{Cov}(\hat{C}_T^{\tau_1}, \hat{C}_T^{\tau_2} \mid \tau_1, \tau_2) = \operatorname{Cov}(\hat{C}_T^{\tau_1}, \hat{C}_T^{\tau_2})$$

since $\operatorname{E}(C_T^{\tau_1} \mid \tau_1, \tau_2) = C_T$ is independent of τ_1 and τ_2. □

Cross validation

The set of permutations $(\tau_1, \ldots, \tau_k)$ such that

$$\coprod_{i=1}^{k} \tau_i(\{N_{\text{train}} + 1, \ldots, N\}) = \{1, \ldots, N\}$$

where $\coprod$ denotes disjoint union is called the set of *split permutations*. A k-fold cross validation is a k-subsampling procedure with $N_{\text{test}}k = N$, where the joint distribution of the permutations $(\tau_1, \ldots, \tau_k)$ is the discrete uniform distribution with support on the set of split permutations. It is not difficult to see that the joint density of the permutations is exchangeable.

In the above definition of cross validation we assumed that N is a multiple of k and that the number of training samples in each split is $N - \frac{N}{k}$. In many applications this assumption is not met and the number of training samples varies over the splits. Proposition 3 can not trivially be generalized to handle such subsampling procedures since the parameter estimator $\hat{\beta}$ depend on the number of training samples – and a priori there is no connection between estimators with different number of training samples, see definition 10.

By Proposition 3 the variance of the cross validation estimator is for k-fold cross validation

$$\operatorname{Var}(\hat{C}_T) = \frac{1}{N}\operatorname{E}(\sigma^2) + \frac{1}{k}\operatorname{Var}(\mu) + \left(1 - \frac{1}{k}\right)\operatorname{Cov}(\hat{C}_T^{\tau_1}, \hat{C}_T^{\tau_2}).$$

Assume that the estimator $\hat{\beta}$ is constant under the cross validation permutations of the data, that is we may take $\hat{\beta}$ to be independent of τ. Then the variance is

$$\operatorname{Var}(\hat{C}_T) = \frac{1}{N}\sigma^2$$

since $\operatorname{Var}(\mu) = 0$ and $\operatorname{Cov}(\hat{C}_T^{\tau_1}, \hat{C}_T^{\tau_2}) = 0$. Note that stability of $\hat{\beta}$ and independence of $\mathfrak{D}_{\text{train}}^{\tau_1}$ and $\mathfrak{D}_{\text{test}}^{\tau_2}$ ensures $\operatorname{Cov}(\hat{C}_T^{\tau_1}, \hat{C}_T^{\tau_2}) = 0$. Hence we can conclude that if the estimator is stable under the cross validation permutations then the variance is independent of k, this is an old result see Kohavi [8].

The assumption of stability of the estimator seldom holds in practice, although we see from the example of Figure 3.6 that for $k \leq 20$ the main contribution to the variance does come from term $\frac{1}{N}\operatorname{E}(\sigma^2)$. It is also interesting that leave one out cross validation ($k = 40$) has the highest variance, which is seen to be due to the covariance term being quite large.

Standard subsampling

The standard M subsampling procedure is a M-subsampling procedure with the joint distribution of the permutations $(\tau_1, \ldots, \tau_M)$ the discrete uniform distribution. That is for standard M subsampling the permutations $\tau_1, \ldots, \tau_M$ are i.i.d.. For a *standard subsampling* procedure we typically choose M fairly high, say $M \leq 100$, this implies that

$$\operatorname{Var}(\hat{C}_T) \approx \operatorname{Cov}(\hat{C}_T^{\tau_1}, \hat{C}_T^{\tau_1}).$$

This result is also evident from the example in Figure 3.6.

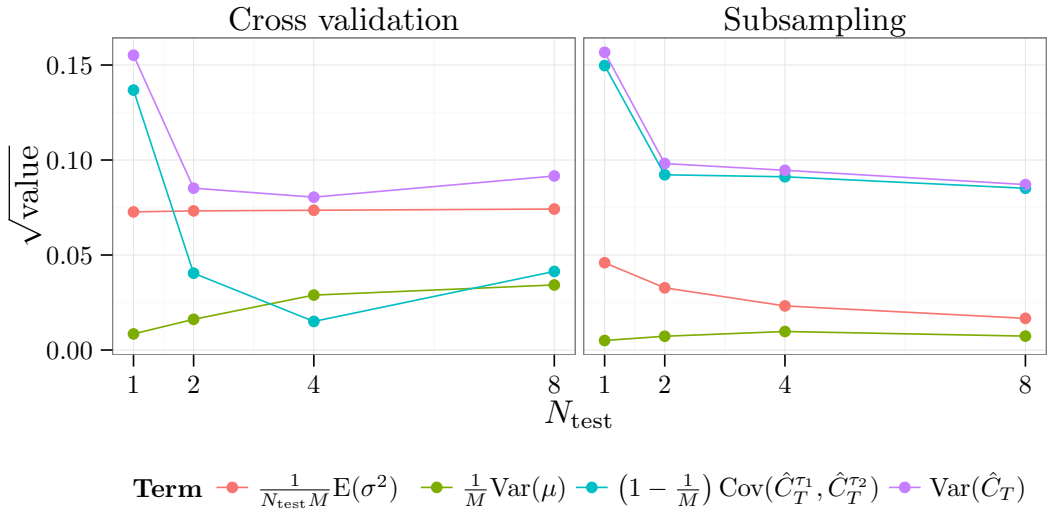


Figure 3.6: The different contributions, on a simulated data set, to the variance of Cross validation estimates and standard subsampling estimates of the expected 01 loss. The values of the different contributions are shown as a function of the number of test samples N_{test} in each subsample. The data set was simulated using the simulation scheme of section 2.5 and with the *Childhood Leukemia* data set as template. For the Cross validation the fold k is N/N_{test}, for the standard subsampling $M = 100$. In all cases the total number of samples N were 40. In order to estimate the contributions of the different terms 200 data sets were simulated, all drawn from the same distribution. This distribution were of the form (2.10), with the Bayes classifier obtained using multinomial group lasso. For each of the 200 data sets two models were estimated, corresponding to τ_1 and τ_2, using group lasso. Furthermore for each of the 200 data sets the test errors $\hat{C}_T^{\tau_1}$ and $\hat{C}_T^{\tau_2}$ were computed and the true error and true variance were computed using formula (2.11). The contributions of the terms were then estimated as the corresponding empirical statistics.

Chapter 4

Sublinear penalization

4.1 Introduction

In this chapter we discuss a generalization of the lasso, group lasso and sparse group lasso penalties. We will consider penalized empirical risk minimization with *sublinear penalties*, which includes sparse group lasso penalties and much more. A penalty is sublinear if and only if it is convex and positively homogeneous, see Appendix C for an equivalent definition in terms of *sub-linearity*. This may seem like a very broad generalization, however, it is possible to derive optimality conditions for the solutions, an exact solution for quadratic empirical risk and generic algorithms for such optimization problems. We will introduce the concept of *decomposition* of a penalty, that will allow us to decompose the optimality conditions into a collection of simpler conditions. Moreover a decomposition also allow us to solve the penalized optimization problem by using block coordinate descent methods, that is by sequentially solving simpler optimization problems.

Throughout this chapter we will use several results from convex analysis, Urruty and Lemaréchal [18] covers everything we need except coordinate descent methods. A short review tailored to this chapter is given in appendix B. The result regarding coordinate decent for non-differentiable optimization, which we need, can be found in Tseng and Mangasarian [16] and Tseng and Yun [17]. A short review is given in appendix A of Vincent and Hansen [21]. For a general introduction to convex optimization see for example Boyd and Vandenberghe [3].

4.2 The penalty

In this section we motivate the definition of sublinear penalties and define the concept of decomposition of a penalty. In order to do this we assume, as in section 3.2, that the parameters are grouped. That is we decompose the parameter space

$$\mathbb{R}^n = \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_m}$$

into $m \in \mathbb{N}$ groups. The groups having dimensions $n_i \in \mathbb{N}$ for $i = 1, \ldots, m$, hence $n = n_1 + \cdots + n_m$. For a vector $\beta \in \mathbb{R}^n$ we write $\beta = (\beta^{(1)}, \ldots, \beta^{(m)})$ where $\beta^{(1)} \in \mathbb{R}^{n_1}, \ldots, \beta^{(m)} \in \mathbb{R}^{n_m}$. For $J = 1, \ldots, m$ we call $\beta^{(J)}$ the J'th *group* of β. We use the notation $\beta_i^{(J)}$ to denote the i'th coordinate of the J'th group of β, whereas β_i is the i'th coordinate of β.

The penalized risk minimization estimator is defined as a solution to the optimization problem

$$\underset{\beta \in B}{\text{minimize}}\, \hat{R}_D(\beta) + \lambda \Phi(\beta) \tag{4.1}$$

with $\hat{R}_D$ the empirical risk, $\Phi : \mathbb{R}^p \to \mathbb{R}$ the penalty and with $\lambda > 0$. All the penalties that we will consider are convex, hence if the empirical risk $\hat{R}_D$ is convex then the estimator $\hat{\beta}$ is a solution to a convex optimization problem.

We are primarily interested in separable sublinear penalties, the separability of the penalty implies that the optimization problem (4.1) separates into a collection of optimization problems. With each of these problems having a lower dimension than the primary problem (4.1). Separability means that Φ is a sum of sublinear functions – a function is sublinear if and only if it is convex and positively homogeneous, see Appendix B – in the following way

$$\Phi(\beta) = \sum_{J=1}^{m} \sigma_J(\beta^{(J)})$$

where $\sigma_J : \mathbb{R}^{n_J} \to \mathbb{R}$ is sublinear.

Sublinear functions are in bijective correspondence with support functions of compact convex sets, this implies that there exists m nonempty compact convex sets $C_1 \subseteq \mathbb{R}^{n_1}, \ldots, C_m \subseteq \mathbb{R}^{n_m}$ such that

$$\Phi(\beta) = \sum_{J=1}^{m} \sigma_{C_J}(\beta^{(J)})$$

where σ_{C_J} denotes the support function of C_J. If we by $\iota_J : \mathbb{R}^{n_J} \hookrightarrow \mathbb{R}^p$ denote the canonical inclusion into the subspace spanned by the J'th parameter group, then by (B.5)

$$\sigma_{C_J}(\beta^{(J)}) = \sigma_{\iota_J(C_J)}(\iota_J(\beta^{(J)})).$$

Hence there exists m nonempty compact convex sets $C_1 \subseteq \mathbb{R}^p, \ldots, C_m \subseteq \mathbb{R}^p$ such that

$$\Phi(\beta) = \sum_{J=1}^{m} \sigma_{C_J}(P_J \beta) \tag{4.2}$$

where $P_J : \mathbb{R}^p \to \mathbb{R}^p$ is the projection onto the subspace spanned by the J'th parameter group. Furthermore the sets $C_1, \ldots, C_m$ can be chosen such that $C_J = P_J C_J$ for all $J = 1, \ldots, m$.

For a support function σ_C we have for $\lambda > 0$ the relation $\lambda \sigma_C = \sigma_{\lambda C}$. This implies that we without loss of generality may consider minimizers of

$$\hat{R}_D(\beta) + \Phi(\beta).$$

The interesting penalties are the non-differentiable ones – since these penalties induce *feature selection* properties. We can therefore not restrict our attention to differentiable optimization problems of the form (4.1). Therefore, in order to better understand the solutions of (4.1) we must use subdifferential calculus – instead of ordinary differential calculus – to derive optimality conditions. The subdifferential generalizes the ordinary gradient, see for example Urruty and Lemaréchal [18]. In the next section we will use the subdifferential calculus to obtain optimality conditions for (4.1).

Example 6 (Norms)**.** *The dual norm* $\|\cdot\|^*$ *of a norm* $\|\cdot\|$ *is*

$$\|z\|^* \stackrel{def}{=} \sup\{z^T v \mid \|v\| \leq 1\}.$$

Furthermore it can be shown that the dual of the dual norm is the original norm. It follows that if U^* denotes the unit ball of the dual norm then

$$\|x\| = \sigma_{U^*}(x).$$

This implies that we may write the canonical lasso penalty, see example 4, as

$$\|\beta\|_1 = \sigma_{[-1,1]^{Kp}}(\beta) = \sum_{i=1}^{K}\sum_{j=1}^{p}\sigma_{[-1,1]}(\beta_{ij})$$

since the dual of the 1-norm is the ∞-norm. Since the 2-norm is self dual the canonical group lasso penalty, see example 5, may be written as

$$\sum_{I=1}^{p}\left\|\beta^{(I)}\right\|_2 = \sum_{I=1}^{p}\sigma_{U_2}(\beta^{(I)})$$

where U_2 is the unit ball of the 2-norm.

4.2.1 Sublinear penalty

The penalty Φ discussed above is itself sublinear and by grouping all parameters together any sublinear function can be taken as a penalty. By the bijective correspondence between sublinear functions and support functions of compact convex sets if follows that for any sublinear penalty $\Phi : \mathbb{R}^p \to \mathbb{R}$ there exist a compact convex set C such that $\Phi = \sigma_C$.

When dealing with high dimensional problems it is essential that the penalty is separable. We will use a slightly broader notion than separability as discussed above, we shall say that a penalty is *decomposable* if there exists a non-trivial *decomposition* of the penalty. Where we define:

Definition 18. *A decomposition of the penalty Φ is a collection, $P_1, \ldots, P_m$ of projections on $\mathbb{R}^p$, such that the following two conditions are fulfilled*

1. *The linear map $\sum_{J=1}^{m} P_J$ is the identity.*
2. *The collection decomposes C, that is the set equality $C = P_1C + \cdots + P_mC$ holds.*

There will in general be multiple ways to decompose a penalty. The idea is that a decomposition breaks the large optimization problem (4.1) into smaller problems which are easier to solve. And by solving these problems sequentially a solution to the complete problem can be obtained, we will discuss this further in section 4.5.

Given a grouping of the parameters and a penalty as defined in terms of (4.2) we may define a sublinear penalty by setting $\Phi(\beta) = \sigma_C(\beta)$ where $C = C_1 + \cdots + C_m$. And a decomposition by letting P_J be the projection onto the subspace of $\mathbb{R}^p$ spanned by the J'th group of parameters. By lemma 4 below the two definitions of Φ agree.

Lemma 4. *Given a decomposition of Φ let $C_J = P_JC$ for $J = 1, \ldots, m$, then*

$$\Phi(\beta) = \sum_{J=1}^{m}\sigma_{C_J}(P_J\beta).$$

Proof. Since the collection $P_1, \ldots, P_m$ of projections decompose C it follows that

$$\Phi(\beta) = \sigma_C(\beta) = \sum_{J=1}^{m} \sigma_{C_J}(\beta).$$

Furthermore since P_J is symmetric $\sigma_{P_J C_J}(\beta) = \sigma_{C_J}(P_J\beta)$ for all $J = 1, \ldots, m$, hence

$$\begin{aligned}\sum_{J=1}^{m} \sigma_{C_J}(\beta) &= \sum_{J=1}^{m} \sigma_{P_J C_J}(\beta) \\ &= \sum_{J=1}^{m} \sigma_{C_J}(P_J\beta).\end{aligned}$$

□

4.3 Optimality conditions

Optimality conditions for (4.1) can be obtained using the subdifferential, as we will do in Theorem 3 below. A decomposition of Φ separates the optimality conditions into parts. The following theorem states a necessary and sufficient optimality condition:

Theorem 3. *Given a decomposition $P_1, \ldots, P_m$ of Φ, the vector $\hat{\beta} \in \mathbb{R}^p$ is a solution to (4.1) if and only if the following two conditions are fulfilled for all $I = 1, \ldots, m$*

1. $-P_I \nabla \hat{R}_D(\hat{\beta}) \in \lambda C_I$
2. $P_I\hat{\beta} \in \lambda N_{C_I}\left(-\frac{1}{\lambda} P_I \nabla \hat{R}_D(\hat{\beta})\right)$ *when* $P_I\hat{\beta} \neq 0$.

Where $N_C(x)$ is the *normal cone* to C at x, see Appendix B. A proof of the Theorem will be given at the end of this section. Figure 4.1 illustrates the Theorem for the ℓ_2-norm (group lasso) and ℓ_1-norm (lasso) penalty using a group decomposition. The Theorem has two important Corollaries. First, when a group of parameters is nonzero then the corresponding gradient lies on the boundary of λC_I, that is:

Corollary 2. *Let $\hat{\beta}$ be a solution to (4.1). If $P_I\hat{\beta} \neq 0$ then $-P_I \nabla \hat{R}_D(\hat{\beta})$ lies on the boundary of λC_I.*

Second, a necessary and sufficient condition that 0 is a solution:

Corollary 3. *The zero vector is a solution to (4.1) if and only if $\nabla \hat{R}_D(0) \in \lambda C$.*

The importance of Corollary 3 is seen when used in connection with a block coordinate descent method, then the Corollary may be used as a computationally efficient way to check if a group of parameters is 0. Algorithms are discussed further in section 4.5. It is clear from Corollary 3 that for sufficiently large values of λ the zero vector is a solution to (4.1). The infimum of these is denoted $\lambda_{\max}$. Given a decomposition of Φ and by Corollary 3 it follows that

$$\begin{aligned}\lambda_{\max} &\overset{\text{def}}{=} \inf\{\lambda > 0 \mid \text{zero is a solution to (4.1) }\} \\ &= \inf\{\lambda > 0 \mid \text{for all } I = 1, \ldots, m \text{ it holds that } -P_I \nabla \hat{R}_D(0) \in \lambda C_I\} \\ &= \max_{I=1,\ldots,m} \inf\{\lambda > 0 \mid -P_I \nabla \hat{R}_D(0) \in \lambda C_I\}.\end{aligned}$$

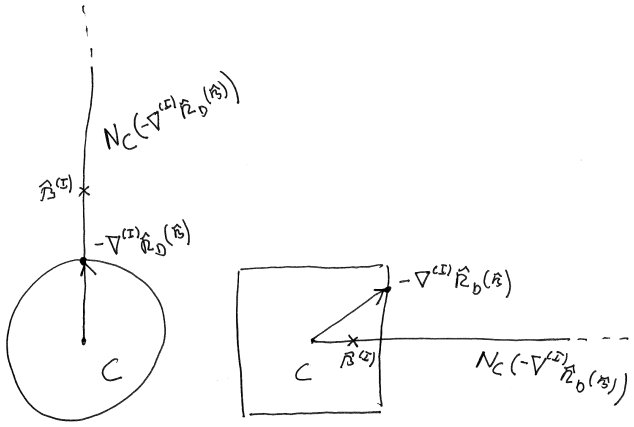


Figure 4.1: Illustration of the optimality condition for respectively the ℓ_2-norm (group lasso) and ℓ_1-norm (lasso) penalty.

4.3.1 Group decomposition

In most cases we are interested in decompositions of Φ that reflect a grouping of the variables. We shall use the following group notation for the gradient: For $I = 1, \ldots, m$ the notation $\nabla^{(I)} \hat{R}_D(\beta)$ stands for the n_I dimensional row vector defined by

$$\nabla \hat{R}_D(\beta) = \left(\nabla^{(1)} \hat{R}_D(\beta), \nabla^{(2)} \hat{R}_D(\beta), \ldots, \nabla^{(m)} \hat{R}_D(\beta) \right).$$

When a decomposition reflects a grouping of the parameters then, by Theorem 3, $\hat{\beta}$ is a solution of (4.1) if and only if;

1. $-\nabla^{(I)} \hat{R}_D(\hat{\beta}) \in \lambda C_I$
2. $\hat{\beta}^{(I)} \in \lambda N_{C_I} \left(-\frac{1}{\lambda} \nabla^{(I)} \hat{R}_D(\hat{\beta}) \right)$ when $\hat{\beta}^{(I)} \neq 0$

for all $I = 1, \ldots, m$.

4.3.2 Proof of Theorem 3

Proof. By convexity of $\hat{R}_D$ and the penalty it follows that $\hat{R}_D(\hat{\beta}) + \lambda\Phi(\hat{\beta})$ is convex. This implies that $\hat{\beta}$ is a minimizer if and only if

$$0 \in \nabla \hat{R}_D(\hat{\beta}) + \lambda \partial \Phi(\hat{\beta})$$

where $\partial\Phi$ denotes the subdifferential of Φ.

Using the decomposition of Φ and that for a support function the subdifferential $\partial\sigma_C(x) \subseteq C$ we find that $\hat{\beta}$ is a minimizer if and only if for each $I = 1, \ldots, m$

$$0 \in P_I \nabla \hat{R}_D(\hat{\beta}) + \lambda \partial \sigma_{C_I}(P_I \hat{\beta}). \tag{4.3}$$

The subdifferential at zero of a support function σ_{C_I} is C_I. Furthermore the subdifferential of σ_{C_I} at $P_I\hat{\beta} \neq 0$ is the face of C_I exposed by $P_I\hat{\beta}$ see Urruty and Lemaréchal [18]. Hence $\hat{\beta} \in \mathbb{R}^p$ is a solution to (4.1) if and only if for all $I = 1, \ldots, m$

1. $-P_I\nabla\hat{R}_D(\hat{\beta}) \in \lambda C_I$ when $P_I\hat{\beta} = 0$.
2. $-P_I\nabla\hat{R}_D(\hat{\beta})$ lie in the face of λC_I exposed by $P_I\hat{\beta}$ when $P_I\hat{\beta} \neq 0$.

Condition 2. above is equivalent to $-P_I\nabla\hat{R}_D(\hat{\beta}) \in \lambda C_I$ and

$$P_I\hat{\beta} \in N_{\lambda C_I}\left(-P_I\nabla\hat{R}_D(\hat{\beta})\right).$$

The statement of the Theorem follows by noting that for a convex set C and $x \in C$

$$\begin{aligned} N_{\lambda C}(x) &= \{s \in \mathbb{R}^n \mid s^T(y-x) \leq 0 \text{ for all } y \in \lambda C\} \\ &= \{s \in \mathbb{R}^n \mid (s/\lambda)^T(y - x/\lambda) \leq 0 \text{ for all } y \in C\} \\ &= \lambda N_C\left(\frac{x}{\lambda}\right) \end{aligned}$$

□

4.4 Exact solution for quadratic empirical risk

It this section we will derive an exact solution to the optimization problem (4.1) when the empirical risk is quadratic. Such a solution will give some insight into the working of these methods and can possibly be used to derive properties of the resulting estimator. Moreover the formula may be used to construct efficient algorithms for computing the minimizer. This can be done in connection with the coordinate gradient decent method and a decomposition of Φ, as discussed further in the next section. We assume that $\Phi = \sigma_C$ for some compact convex set C and consider the case when $\hat{R}_D$ is quadratic, i.e.

$$\hat{R}_D(\beta) = q^T\beta + \beta^T H\beta$$

with H (symmetric) positive definite.

An essential observation, that we need to derive a exact solution, is that we can solve the equation

$$-q - H\beta = P_C(-q - H\beta + \beta) \tag{4.4}$$

where $P_C : \mathbb{R}^p \to \mathbb{R}^p$ is the projection onto the set C – see appendix B. The solution is given in the following lemma.

Lemma 5. *For positive definite matrix H the unique solution to equation (4.4) is*

$$\beta = -A^{-1}P_{A^{-1}C}(-A^{-1}q) - H^{-1}q \tag{4.5}$$

where $A = \sqrt{H}$.

Proof. We note that for $y \in C$

$$\begin{aligned} \beta^T(y + q + H\beta) &= \beta^T AA^{-1}(y + q + H\beta) \\ &= (-A^{-1}q + A^{-1}q + A\beta)^T(A^{-1}y + A^{-1}q + A\beta). \end{aligned}$$

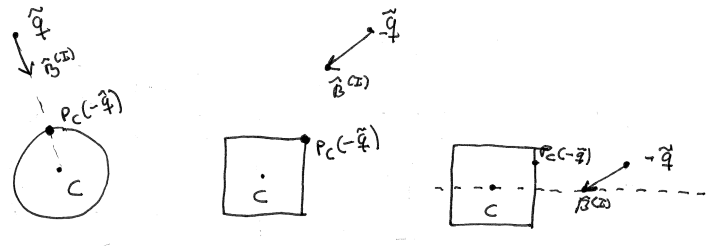


Figure 4.2: Illustration of the solution the optimization problem (4.1) in two dimensions with different penalties; the group lasso (with one group), the lasso with a non sparse solution and the lasso with a sparse solution.

By (B.9) this implies that (4.4) is equivalent to

$$A^{-1}q + A\beta = P_{A^{-1}C}(-A^{-1}q)$$

which in turn is seen to be equivalent to (4.5). □

We are now ready to give the formula for the solution to 4.

Theorem 4. *The minimizer $\hat{\beta}$ of (4.1) is 0 if $-q \in C$, otherwise*

$$\hat{\beta} = -A^{-1}P_{A^{-1}C}(-A^{-1}q) - H^{-1}q.$$

where $A = \sqrt{H}$.

Proof. In order to proof that $\hat{\beta}$ is a solution we must show that condition 1 and 2 of Theorem 3 is fulfilled. The gradient of $\hat{R}_D$ is

$$\nabla \hat{R}_D(\beta) = q + H\beta.$$

If $\hat{\beta} = 0$ then $-\nabla \hat{R}_D(\hat{\beta}) = -q \in C$, hence condition 1 and 2 are fulfilled. So assume that $\hat{\beta} \neq 0$, by lemma 5

$$-q - H\hat{\beta} = P_C(-q - H\hat{\beta} + \hat{\beta}) \tag{4.6}$$

which implies that $-\nabla \hat{R}_D(\hat{\beta}) = -q - H\hat{\beta} \in C$, hence condition 1 is fulfilled. Furthermore by the relation (B.10) between the normal cone to C and the projection onto C equation (4.6) implies that

$$\hat{\beta} \in \lambda N_C(-q - H\hat{\beta}),$$

hence condition 2 is fulfilled. □

If $H = \mathrm{id}$ then $\hat{\beta}^{(I)} = -P_C(-\tilde{q}) - \tilde{q}$ this cases is illustrated in Figure 4.2.

4.5 Algorithms

In this section we present two algorithms that can be used to solve the penalized minimization problem with sublinear penalty. A block coordinate descent algorithm and a coordinate gradient

descent algorithm. For the sparse group lasso problem we used in Vincent and Hansen [21] a slightly different approach, namely a nested coordinate gradient descent.

We assume given a grouping of the parameters and assume that this grouping decompose the penalty Φ. The resulting minimization problem is separable, with respect to this grouping, non-differentiable and convex. Define $\hat{\beta}_{\circ I} \in \mathbb{R}^p$ by setting

$$P_J \hat{\beta}_{\circ I} = P_J \hat{\beta} \text{ for all } J \neq I \text{ and } P_I \hat{\beta}_{\circ I} = 0.$$

The *partial optimization problem* for the I'th group is

$$\arg\min_{x \in \mathbb{R}^{n_I}} \hat{R}_D(\beta_{\circ I} + \iota_I(x)) + \lambda \sigma_{C_I}(x) \tag{4.7}$$

where ι_I denote the canonical inclusion into the I'th parameter group, i.e. $\iota_I(x)^{(I)} = x$ and $\iota_I(x)^{(J)} = 0$ when $J \neq I$. The partial optimization problems are themselves of the form (4.1). It follows, by Corollary 3, that we may determine if zero is a solution by checking if $-\nabla^{(I)} \hat{R}_D(\beta_{\circ I}) \in \lambda C_I$. Algorithm 1 is a block coordinate descent algorithm with this rule added.

Some additional nonstandard routines are needed to complete the algorithms, these routines depend on the description of the convex sets $C_1, \ldots, C_m$. For both the algorithms presented here a routine for checking if a vector is contained in C_I is needed for all $I = 1, \ldots, m$. For the block coordinate descent an additional routine for solving the partial optimization problems (4.7) is needed for all $I = 1, \ldots, m$. For the coordinate gradient descent a routine for solving the convex optimization problem

$$\arg\min_{y \in \lambda C_I} \left\| A^{-1}(y + \tilde{q}) \right\|_2^2$$

with A a positive definite matrix is need for all $I = 1, \ldots, m$ and with $\lambda > 0$.

4.5.1 Block coordinate descent

Block coordinate descent is an iterative method where a sequence of parameters $\{\beta_N\}$ is constructed by sequentially solving each of the m partial optimization problems (4.7). Tseng and Mangasarian [16] showed that, for separable non-differentiable minimization, block coordinate descent converges. This implies that Algorithm 1 will provide us with a sequence converging to a solution of (4.1).

while *until stopping condition is met* **do**
  Choose next block index I according to the cyclic rule.
  if $-\nabla^{(I)} \hat{R}_D(\beta_{old,\circ I}) \in \lambda C_I$ **then**
    Let $\beta^{(I)}_{\text{new}} = 0$.
  else
    Let
    $$\beta^{(I)}_{\text{new}} = \arg\min_{x \in \mathbb{R}^{n_I}} \hat{R}_D(\beta_{\text{old},\circ I} + \iota_I(x)) + \lambda \sigma_{C_I}(x)$$
  end
end

Algorithm 1: Block coordinate descent with rule for checking if a block is zero.

4.5.2 Coordinate gradient descent

Another more complex but usually also more efficient algorithm applicable for sublinear penalized minimization problems is a coordinate gradient descent method. Coordinate gradient descent for separable non-differentiable minimization is addressed in details by Tseng and Yun [17]. Coordinate gradient descent is similar to block coordinate descent except quadratic approximations are being sequentially optimized. The convergence of Algorithm 2 is implied by Theorem 3 and 4, by realizing that it is simply a coordinate gradient descent algorithm.

while *until stopping condition is met* **do**
  Choose next block index I according to the cyclic rule.
  Construct a quadratic approximation

$$Q(\beta) \sim q^T\beta + \frac{1}{2}\beta^T H\beta$$

  of $\hat{R}_D$ near β_{old}.
  Let $\tilde{q} = \nabla^{(I)} Q(\beta_{\text{old},\circ I}) = q^{(I)} + (H\beta_{\text{old},\circ I})^{(I)}$
  if $-\tilde{q} \in \lambda C_I$ **then**
    Let $\beta^{(I)}_{\text{new}} = 0$.
  else
    Let $A = \sqrt{H_{II}}$ and solve the convex optimization problem

$$t = \underset{y \in \lambda C_I}{\arg\min} \left\| A^{-1}(y + \tilde{q}) \right\|_2^2 .$$

    Let $\beta^{(I)}_{\text{new}} = -A^{-1}t - H_{II}^{-1}\tilde{q}$.
  end
end

Algorithm 2: Coordinate gradient descent algorithm. The square matrix H_{II} denotes the diagonal block of the Hessian matrix corresponding to the I'th group

4.6 Multinomial sparse group lasso

In this section the multinomial sparse group lasso is shortly discussed, see Vincent and Hansen [21] for further discussion. The sparse group lasso penalty is defined as

$$\underbrace{\sum_{J=1}^{m} \gamma_J \left\| \beta^{(J)} \right\|_2}_{\text{group lasso}} + \underbrace{\sum_{i=1}^{n} \xi_i \left| \beta_i \right|}_{\text{lasso}} \tag{4.8}$$

for group weights $\gamma \in [0,\infty)^m$, and parameter weights $\xi = (\xi^{(1)}, \ldots, \xi^{(m)}) \in [0,\infty)^n$ where $\xi^{(1)} \in [0,\infty)^{n_1}, \ldots, \xi^{(m)} \in [0,\infty)^{n_m}$. We emphasize that the penalty is specified by

- a grouping of the parameters $\beta = (\beta^{(1)}, \ldots, \beta^{(m)})$,
- and the weights γ and ξ.

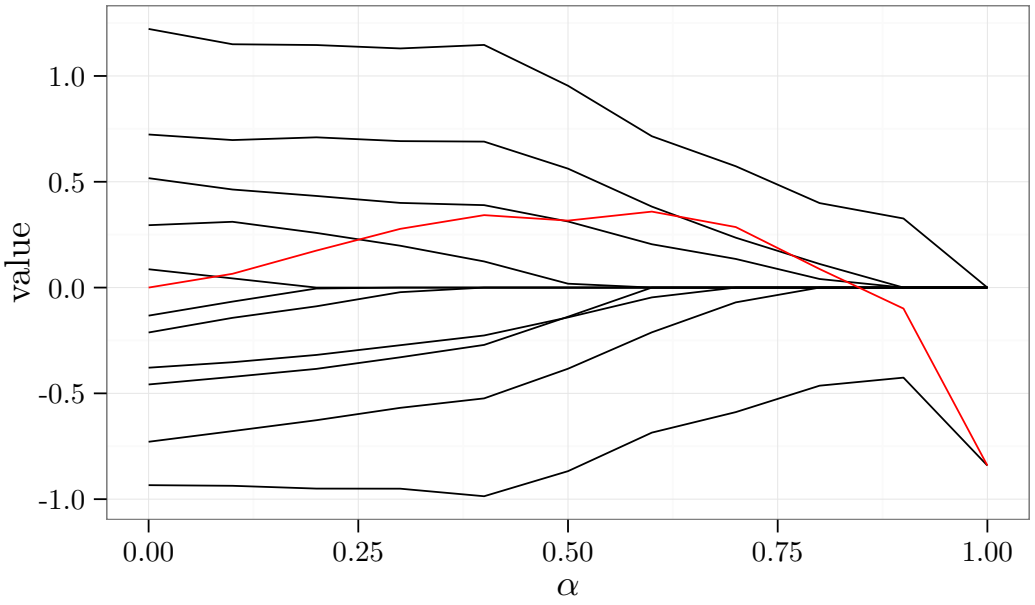


Figure 4.3: Values of the estimated parameters corresponding to one covariate, as a function of α in the sparse group lasso penalty (4.9). The covariate is miR 17 in the *Primary Cancers* data set. Each black line correspond to a parameter, the red line is the sum of the parameters.

We may rewrite the penalty (4.8) as

$$\sum_{J=1}^{m} \left\| \beta^{(J)} \right\|_J$$

where

$$\left\| \beta^{(J)} \right\|_J = \gamma_J \left\| \beta^{(J)} \right\|_2 + \sum_{i=1}^{n_J} \xi_i^{(J)} \left| \beta_i^{(J)} \right|$$

is a norm whenever $\gamma_J \neq 0$. If $\gamma_J = 0$ and the parameter weight at a coordinate i of the J'th block is 0, i.e. $\xi_i^{(J)} = 0$, then the penalty is a semi-norm. The penalty is in particular sublinear. We have in Figure 4.3 plotted the values of the parameters – of the multinomial model – corresponding to a particular covariate, as the penalty is varied from the group lasso to the lasso. This is done by considering the collection of sparse group lasso penalties

$$(1-\alpha) \sum_{J=1}^{m} \sqrt{K} \left\| \beta^{(J)} \right\|_2 + \alpha \sum_{i=1}^{n} |\beta_i| \tag{4.9}$$

parametrized by $\alpha \in [0, 1]$.

If we let $\bigcirc$ denote the unit ball of the 2-norm and $\square$ the unit square i.e. the unit ball of the ∞-norm, then the sparse group lasso penalty $\|\beta\|_2 + |\beta_1| + |\beta_2|$ is equal to $\sigma_{\square} + \sigma_{\bigcirc} = \sigma_{\square + \bigcirc}$. In other words the convex set $\square + \bigcirc$ is associated with the sparse group lasso penalty $\|\beta\|_2 + |\beta_1| + |\beta_2|$. See figure 4.4 for an illustration of the convex set associated with the sparse group lasso. Knowing the associated convex sets of the penalties makes it possible for us to do a geometric comparison

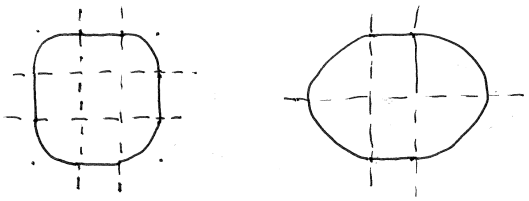


Figure 4.4: The convex sets associated with two different configurations of the sparse group lasso penalty in two dimensions. The penalties are $\|\beta\|_2 + |\beta_1| + |\beta_2|$ and $\|\beta\|_2 + |\beta_1|$ respectively.

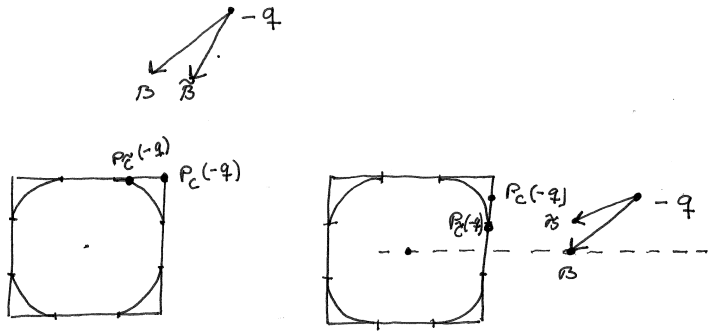


Figure 4.5: Comparison between the sparse group lasso solution $\tilde{\beta}$ and the lasso solution β. The penalties are $\left\|\tilde{\beta}\right\|_2 + \left|\tilde{\beta}_1\right| + \left|\tilde{\beta}_2\right|$ and $2(|\beta_1| + |\beta_2|)$ respectively.

of the solutions using the results of section 4.4. Figure 4.5 is an example of such a geometric comparison between the sparse group lasso and the lasso.

As can be seen on Figure 4.3 the sum of the parameters for the group lasso is zero, this is not a coincidence. For the multinomial group lasso the sum of the estimated parameters within each group will always be 0.

Proposition 4. *For the multinomial group lasso solution $\hat{\beta}$ it holds that*

$$\hat{\beta}_1^{(I)} + \cdots + \hat{\beta}_K^{(I)} = 0$$

for each covariate I.

Proof. For the group lasso the normal cone at $-\frac{1}{\lambda}\nabla^{(I)}\hat{R}_D(\hat{\beta})$ is

$$\{-c\delta \mid c > 0\}$$

with $\delta = \nabla^{(I)}\hat{R}_D(\hat{\beta}) = \frac{1}{N}\sum_{i=1}^{N} x_{iI}(h(\hat{\beta}x_i) - e_{y_i})$. Since

$$h(\hat{\beta}x_i)_1 + \cdots + h(\hat{\beta}x_i) = 1$$

is follows that $\delta_1 + \cdots + \delta_K = 0$. The proposition now follows by Theorem 3. □

Chapter 5

Article: Sparse group lasso

M. Vincent and N. R. Hansen. Sparse group lasso and high dimensional multinomial classification. *Computational Statistics & Data Analysis*, May 2012. URL `http://arxiv.org/abs/1205.1245`

Sparse group lasso and high dimensional multinomial classification.

Martin Vincent[1], Niels Richard Hansen

University of Copenhagen, Department of Mathematical Sciences, Universitetsparken 5, 2100 Copenhagen Ø, Denmark

Abstract

The sparse group lasso optimization problem is solved using a coordinate gradient descent algorithm. The algorithm is applicable to a broad class of convex loss functions. Convergence of the algorithm is established, and the algorithm is used to investigate the performance of the multinomial sparse group lasso classifier. On three different real data examples the multinomial group lasso clearly outperforms multinomial lasso in terms of achieved classification error rate and in terms of including fewer features for the classification. An implementation of the multinomial sparse group lasso algorithm is available in the R package `msgl`. Its performance scales well with the problem size as illustrated by one of the examples considered – a 50 class classification problem with 10k features, which amounts to estimating 500k parameters.

Keywords: Sparse group lasso, classification, high dimensional data analysis, coordinate gradient descent, penalized loss.

1. Introduction

The sparse group lasso is a regularization method that combines the lasso [1] and the group lasso [2]. Friedman et al. [3] proposed a coordinate descent approach for the sparse group lasso optimization problem. Simon et al. [4] used a generalized gradient descent algorithm for the sparse group lasso and considered applications of this method to linear, logistic and Cox regression.

[1]Corresponding author. Tel.: +4522860740
E-mail address: vincent@math.ku.dk (M. Vincent).

We present a sparse group lasso algorithm suitable for high dimensional problems. This algorithm is applicable to a broad class of convex loss functions. In the algorithm we combine three non-differentiable optimization methods: the coordinate gradient descent [5], the block coordinate descent [6] and a modified coordinate descent method.

Our main application is to multiclass classification based on the multinomial regression model. The lasso penalty has, for some time, been considered as a regularization approach for multinomial regression [7]. The parameters in the multinomial model are, however, naturally structured, with multiple parameters corresponding to one feature, and the lasso penalty does not take this structure into account. To accommodate for this we suggest to add a group lasso term with the parameters corresponding to the same feature grouped together. The resulting penalty is known as the sparse group lasso penalty. We found that using the sparse group lasso penalty for multinomial regression generally improved the performance of the estimated classifier and reduced the number of features included in the model.

The formulation of an efficient and robust sparse group lasso algorithm is not straight forward due to non-differentiability of the penalty. Firstly, the sparse group lasso penalty is not completely separable, which is problematic when using a standard coordinate descent scheme. To obtain a robust algorithm an adjustment is necessary. Our solution, which efficiently treats the singularity at zero that cannot be separated out, is a minor modification of the coordinate descent algorithm. Secondly, our algorithm is a Newton type algorithm, hence we sequentially optimize penalized quadratic approximations of the loss function. This approach raises another challenge: how to reduce the costs of computing the Hessian? In Section 3.6 we show that an upper bound on the Hessian is sufficient to determine whether the minimum over a block of coefficients is attained at zero. This approach enables us to update a large percentage of the blocks without computing the complete Hessian. In this way we reduce the run-time, provided that the upper bound of the Hessian can be computed efficiently. We found that this approach reduces the run-time on large data sets by a factor of more than 2.

Our focus is on applications of the multinomial sparse group lasso to problems with many classes. For this purpose we have investigated three multiclass classification problems. We found that multinomial group lasso and sparse group lasso perform well on these problems. The error rates were substantially lower than the best obtained with multinomial lasso, and the low error rates were achieved for models with fewer features having non-zero

coefficients. For example, we consider a text classification problem consisting of Amazon reviews with 50 classes and 10k textual features. This problem showed a large improvement in the error rates: from approximately 40% for the lasso to less than 20% for the group lasso.

We provide a generic implementation of the sparse group lasso algorithm in the form of a `C++` template library. The implementation for multinomial and logistic sparse group lasso regression is available as an R package. For our implementation the time to compute the sparse group lasso solution is of the same order of magnitude as the time required for the multinomial lasso algorithm as implemented in the R package glmnet. The computation time of our implementation scales well with the problem size.

1.1. Sparse group lasso

Consider a convex, bounded below and twice continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$. We say that $\hat{\beta} \in \mathbb{R}^n$ is a *sparse group lasso minimizer* if it is a solution to the unconstrained convex optimization problem

$$\text{minimize } f + \lambda\Phi \tag{1}$$

where $\Phi : \mathbb{R}^n \to \mathbb{R}$ is the *sparse group lasso penalty* (defined below) and $\lambda > 0$.

Before defining the sparse group lasso penalty some notation is needed. We decompose the search space

$$\mathbb{R}^n = \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_m}$$

into $m \in \mathbb{N}$ blocks having dimensions $n_i \in \mathbb{N}$ for $i = 1, \ldots, m$, hence $n = n_1 + \cdots + n_m$. For a vector $\beta \in \mathbb{R}^n$ we write $\beta = (\beta^{(1)}, \ldots, \beta^{(m)})$ where $\beta^{(1)} \in \mathbb{R}^{n_1}, \ldots, \beta^{(m)} \in \mathbb{R}^{n_m}$. For $J = 1, \ldots, m$ we call $\beta^{(J)}$ the J'th *block* of β. We use the notation $\beta_i^{(J)}$ to denote the i'th coordinate of the J'th block of β, whereas β_i is the i'th coordinate of β.

Definition 1 (Sparse group lasso penalty)**.** *The sparse group lasso penalty is defined as*

$$\Phi(\beta) \stackrel{def}{=} (1-\alpha)\sum_{J=1}^{m} \gamma_J \left\|\beta^{(J)}\right\|_2 + \alpha \sum_{i=1}^{n} \xi_i \left|\beta_i\right|$$

for $\alpha \in [0, 1]$, group weights $\gamma \in [0, \infty)^m$, and parameter weights $\xi = (\xi^{(1)}, \ldots, \xi^{(m)}) \in [0, \infty)^n$ where $\xi^{(1)} \in [0, \infty)^{n_1}, \ldots, \xi^{(m)} \in [0, \infty)^{n_m}$.

The sparse group lasso penalty includes the lasso penalty ($\alpha = 1$) and the group lasso penalty ($\alpha = 0$). Note also that for sufficiently large values of λ the solution of (1) is zero. The infimum of these, denoted $\lambda_{\max}$, is computable, see Section 3.2.

We emphasize that the sparse group lasso penalty is specified by

- a grouping of the parameters $\beta = (\beta^{(1)}, \ldots, \beta^{(m)})$,
- and the weights α, γ and ξ.

It is well known that the lasso penalty results in sparse solutions to (1), while the group lasso penalty results in groupwise sparse solutions (that is, the entire group of parameters is zero or non-zero). However group lasso does not give sparsity within groups – sparse group lasso does.

In the second part of the paper we develop an algorithm for solving the optimization problem (1). The convergence of the algorithm is established for any sparse group lasso penalty, regardless of how the parameters are grouped. For multinomial regression, as considered in the next section, we restrict attention to a specific grouping of the parameters that reflects the features. In the symmetric parametrization of the multinomial regression model with K classes there are K parameters per feature. Our suggestion is to group these K parameters together. Thus we do not group the features, only the parameters associated with each feature. For the examples we considered this particular grouping resulted in models with fewer features having non-zero parameters compared to ordinary lasso penalization. More importantly, the error rates were typically also smaller.

Our `msgl` R package supports the particular grouping for multinomial regression as well as additional groupings of the features, i.e. the number of parameters in each group is a multiple of K. The `sgl C++` template library can be configured to handle any grouping.

2. The multinomial sparse group lasso classifier

In this section we examine the characteristics of the multinomial sparse group lasso method. Our main interest is the application of the multinomial sparse group lasso classifier to problems with many classes. For this purpose we have chosen three classification problems based on three different data sets, with 10, 18 and 50 classes. In [8] the microRNA expression profile of different types of primary cancer samples is studied. In Section 2.2.1 we

consider the problem of classifying the primary site based on the microRNA profiles in this data set. The Amazon reviews author classification problem, presented in [9], is studied in Section 2.2.2. The messenger RNA profile of different human muscle diseases is studied in [10]. We consider, in Section 2.2.3, the problem of classifying the disease based on the messenger RNA profiles in this data set. Table 1 summarizes the dimensions and characteristics of the data sets and the associated classification problems. Finally, in Section 2.3, we examine the characteristics of the method applied to simulated data sets.

2.1. Setup

Consider a classification problem with K classes, N samples, and p features. Assume given a data set $(x_1, y_1), \ldots, (x_N, y_N)$ where, for all $i = 1, \ldots, N$, $x_i \in \mathbb{R}^p$ is the observed feature vector and $y_i \in \{1, \ldots, K\}$ is the categorical response. We organize the feature vectors in the $N \times p$ *design matrix*

$$X \stackrel{\text{def}}{=} (x_1 \cdots x_N)^T.$$

As in [7] we use a symmetric parametrization of the multinomial model. With $h : \{1, \ldots, K\} \times \mathbb{R}^p \to \mathbb{R}$ given by

$$h(l, \eta) \stackrel{\text{def}}{=} \frac{\exp(\eta_l)}{\sum_{k=1}^{K} \exp(\eta_k)},$$

the multinomial model is specified by

$$P(y_i = l | x_i) = h(l, \beta^{(0)} + \beta x_i).$$

The model parameters are organized in the K-dimensional vector, $\beta^{(0)}$, of intercept parameters together with the $K \times p$ matrix

$$\beta \stackrel{\text{def}}{=} \left(\beta^{(1)} \cdots \beta^{(p)}\right), \tag{2}$$

where $\beta^{(i)} \in \mathbb{R}^K$ are the parameters associated with the i'th feature.

The log-likelihood is

$$\ell(\beta^{(0)}, \beta) = \sum_{i=1}^{N} \log h(y_i, \beta^{(0)} + \beta x_i). \tag{3}$$

Our interest is the sparse group lasso penalized maximum likelihood estimator. That is, $(\beta^{(0)}, \beta)$ is estimated as a minimizer of the sparse group lasso penalized negative-log-likelihood:

$$-\ell(\beta^{(0)}, \beta) + \lambda \left((1-\alpha) \sum_{J=1}^{p} \gamma_J \left\| \beta^{(J)} \right\|_2 + \alpha \sum_{i=1}^{Kp} \xi_i \left| \beta_i \right| \right). \tag{4}$$

In our applications we let $\gamma_J = \sqrt{K}$ for all $J = 1, \ldots, p$ and $\xi_i = 1$ for all $i = 1, \ldots, Kp$, but other choices are possible in the implementation. Note that the parameter grouping, as part of the penalty specification, is given in terms of the columns in (2), i.e. $m = p$.

A common parametrization of the multinomial regression model singles out a reference class, and the probabilities of the other classes are then given relative to the reference class. As pointed out in [11] this is problematic when lasso penalization is used for parameter estimation, and the symmetric parametrization introduced above, and used in [7] as well, is preferred. It ensures that the resulting estimator is invariant to permutations of the classes. The parameters in the symmetric parametrization are, however, not identifiable. If β_l denotes the l'th row of the matrix β, then for $l, k = 1, \ldots, K$

$$\frac{P(y_i = l|x)}{P(y_i = k|x)} = \exp(\beta_l^{(0)} - \beta_k^{(0)} + (\beta_l - \beta_k)x),$$

and it follows that the differences $\beta_l - \beta_k$ and $\beta_l^{(0)} - \beta_k^{(0)}$ are identifiable. In practice, as was also noted in Section 4.1 in [7], the consequence of the penalization is that the estimated parameters minimize the sparse group lasso penalty among all equivalent parameters. If some parameters, like $\beta^{(0)}$, are not penalized, a procedure like mean centering suggested in [7] can be used to numerically select one of the equivalent parameters.

2.2. Data examples

The data sets were preprocessed before applying the multinomial sparse group lasso estimator. Two preprocessing schemes were used: *normalization* and *standardization*. Normalization is sample centering and scaling in order to obtain a design matrix with row means 0 and row variances 1. Standardization is feature centering and scaling in order to obtain a design matrix with column means 0 and column variances 1. Note that the order in which normalization and standardization are applied matters.

Data set	Features	K	N	p
Cancer sites	microRNA expressions	18	162	217
Amazon reviews	Various textual features	50	1500	10k
Muscle diseases	Gene expression	10	107	22k

Table 1: Summary of data sets and the associated classification problem.

The purpose of normalization is to remove technical (non-biological) variation. A range of different normalization procedures exist for biological data. Sample centering and scaling is one of the simpler procedures. We use this simple normalization procedure for the two biological data sets in this paper. Normalization is done before and independently of the sparse group lasso algorithm.

The purpose of standardization is to create a common scale for the features. This ensures that differences in scale will not influence the penalty and thus the variable selection. Standardization is an option for the sparse group lasso implementation, and it is applied as the last preprocessing step for all three example data sets.

We want to compare the performance of the multinomial sparse group lasso estimator for different values of the regularization parameter α. Applying the multinomial sparse group lasso estimator with a given $\alpha \in [0, 1]$ and λ-sequence, $\lambda_1, \ldots, \lambda_d > 0$, results in a sequence of estimated models with parameters $\{\hat{\beta}(\lambda_i, \alpha)\}_{i=1,\ldots,d}$. The generalization error can be estimated by cross validation [12]. For our applications we keep the sample ratio between classes in the cross validation subsets approximately fixed to that of the entire data set. Hence, we may compute a sequence, $\{\widehat{\mathrm{Err}}(\lambda_i, \alpha)\}_{i=1,\ldots,d}$, of estimated expected generalization errors for the sequence of models. However, for given α_1 and α_2 we cannot simply compare $\widehat{\mathrm{Err}}(\lambda_i, \alpha_1)$ and $\widehat{\mathrm{Err}}(\lambda_i, \alpha_2)$, since the λ_i value is scaled differently for different values of α. We will instead compare the models with the same number of non-zero parameters and the same number of non-zero parameter groups, respectively. Define

$$\hat{\Theta}(\lambda, \alpha) \stackrel{\text{def}}{=} \sum_{J=1}^{p} I(\hat{\beta}^{(J)}(\lambda, \alpha) \neq 0)$$

with $\hat{\beta}(\lambda, \alpha)$ the estimator of β for the given values of λ and α. That is, $\hat{\Theta}(\lambda, \alpha)$ is the number of non-zero parameter blocks in the fitted model.

Note that there is a one-to-one correspondence between parameter blocks and features in the design matrix. Furthermore, we define the total number of non-zero parameters as

$$\hat{\Pi}(\lambda, \alpha) \stackrel{\text{def}}{=} \sum_{i=1}^{n} I(\hat{\beta}_i(\lambda, \alpha) \neq 0).$$

In particular, we want to compare the fitted models with the same number of parameter blocks. There may, however, be more than one λ-value corresponding to a given value of $\hat{\Theta}$. Thus we compare the models on a subsequence of the λ-sequence. This subsequence is defined below. With $\theta_1 < \cdots < \theta_{d'}$ for $d' \leq d$ denoting the different elements of the set $\{\hat{\Theta}(\lambda_i, \alpha)\}_{i=1,\ldots,d}$ in increasing order we define

$$\tilde{\lambda}_i(\alpha) \stackrel{\text{def}}{=} \min \left\{ \lambda \;\middle|\; \hat{\Theta}(\lambda, \alpha) = \theta_i \right\}.$$

We then compare the characteristics of the multinomial sparse group lasso estimators for different α values by comparing the estimates

$$\left\{ \left(\widehat{\mathrm{Err}}(\tilde{\lambda}_i(\alpha), \alpha), \hat{\Theta}(\tilde{\lambda}_i(\alpha)), \hat{\Pi}(\tilde{\lambda}_i(\alpha)) \right) \right\}_{i=1,\ldots,d'}.$$

2.2.1. Cancer sites

The data set consists of bead-based expression data for 217 microRNAs from normal and cancer tissue samples. The samples are divided into 11 normal classes, 16 tumor classes and 8 tumor cell line classes. For the purpose of this study we select the normal and tumor classes with more than 5 samples. This results in an 18 class data set with 162 samples. The data set is unbalanced, with the number of samples in each class ranging from 5 to 26 and with an average of 9 samples per class. Data was normalized and then standardized before running the sparse group lasso algorithm. For more information about this data set see [8]. The data set is available from the Gene Expression Omnibus with accession number GSE2564.

Figure 1 shows the result of a 10-fold cross validation for 5 different values of α, including the lasso and group lasso. The λ-sequence runs from $\lambda_{\max}$ to 10^{-4}, with $d = 200$. It is evident that the group lasso and sparse group lasso models achieve a lower expected error using fewer genes than the lasso model. However, models with a low α value have a larger number of non-zero parameters than models with a high α value. A reasonable compromise could

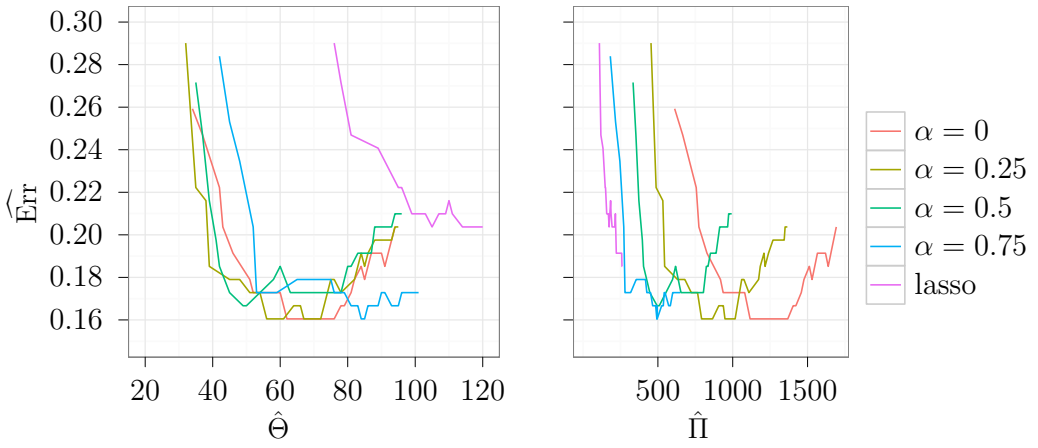


Figure 1: Estimated expected generalization error, for different values of α, for the microRNA cancer sites data set. The cross validation based estimate of the expected misclassification error is plotted against the number of non-zero parameter blocks in the model (left), and against the number of non-zero parameters in the model (right). The estimated standard error is approximately 0.03 for all models.

be the model with $\alpha = 0.25$. This model does not only have a low estimated expected error, but the low error is also achieved with a lower estimated number of non-zero parameters, compared to group lasso.

2.2.2. Amazon reviews

The Amazon review data set consists of 10k textual features (including lexical, syntactic, idiosyncratic and content features) extracted from 1500 customer reviews from the Amazon Commerce Website. The reviews were collected among the reviews from 50 authors with 50 reviews per author. The primary classification task is to identify the author based on the textual features. The data and feature set were presented in [9] and can be found in the UCI machine learning repository [13]. In [9] a Synergetic Neural Network is used for author classification, and a 2k feature based 10-fold CV accuracy of 0.805 is reported. The feature selection and training of the classifier were done separately.

We did 10-fold cross validation using multinomial sparse group lasso for five different values of α. The results are shown in Figure 2. The λ-sequence runs from $\lambda_{\max}$ to 10^{-4}, with $d = 100$. The design matrix is sparse for

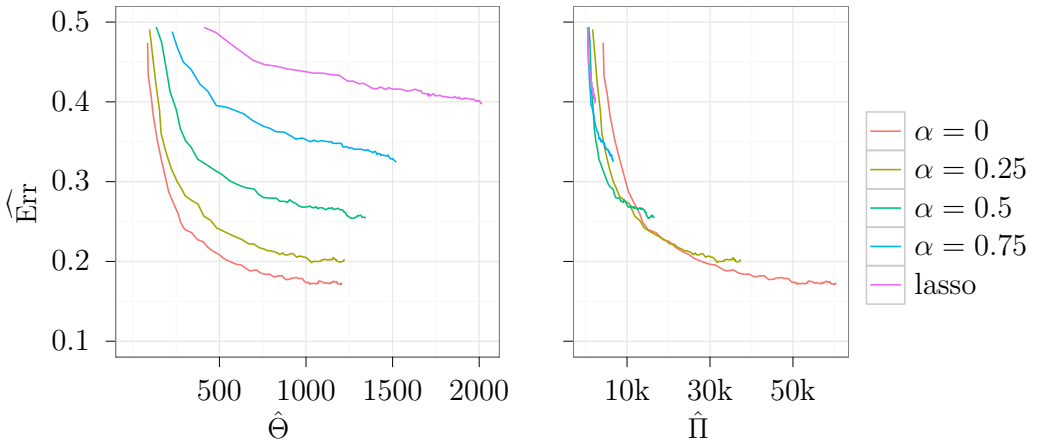


Figure 2: Estimated expected generalization error, for different values of α, for the Amazon reviews author classification problem. The cross validation based estimate of expected misclassification error is plotted against the number of non-zero parameter blocks in the model (left), and against the number of non-zero parameters in the model (right). The estimated standard error is approximately 0.01 for all models.

this data set. Our implementation of the multinomial sparse group lasso algorithm utilizes the sparse design matrix to gain speed and for memory efficiency. No normalization was applied for this data set. Features were scaled to have variance 1, but were not centered.

For this data set it is evident that lasso performs badly, and that the group lasso performs best - in fact much better than lasso. The group lasso achieves an accuracy of around 0.82 with a feature set of size $\sim$ 1k. This outperforms the neural network in [9].

2.2.3. Muscle diseases

This data set consists of messenger RNA array expression data of 119 muscle biopsies from patients with various muscle diseases. The samples are divided into 13 diagnostic groups. For this study we only consider classes with more than 5 samples. This results in a classification problem with 107 samples and 10 classes. The data set is unbalanced with class sizes ranging from 4 to 20 samples per class. Data was normalized and then standardized before running the sparse group lasso algorithm. For background information on this data set, see [10]. The data set is available from the Gene Expression

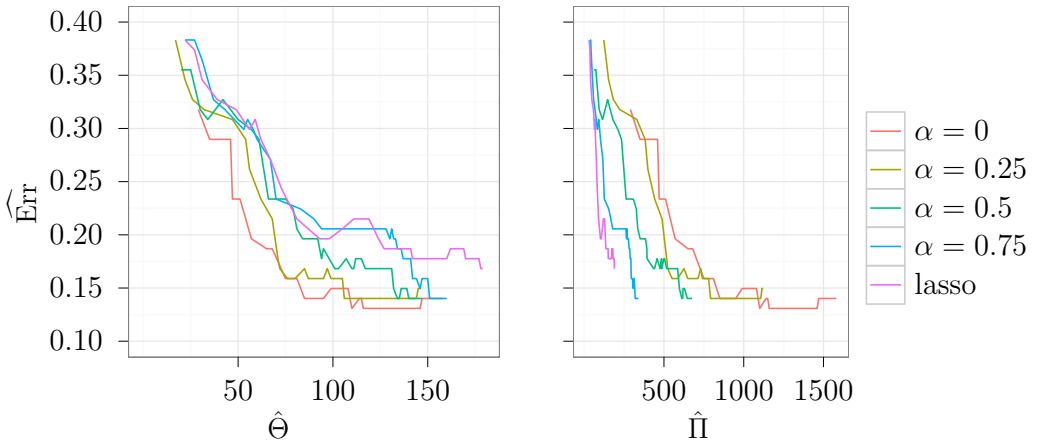


Figure 3: Estimated expected generalization error, for different values of α, for the muscle disease classification problem. The cross validation based estimate of expected misclassification error is plotted against the number of non-zero parameter blocks in the model (left), and against the number of non-zero parameters in the model (right) The estimated standard error is approximately 0.04 for all models.

Omnibus with accession number GDS1956.

The results of a 10-fold cross validation are shown in Figure 3. The λ-sequence runs from $\lambda_{\max}$ to 10^{-5}, with $d = 200$. We see the same trend as in the other two data examples. Again the group lasso models perform well, but not significantly better than the closest sparse group lasso models ($\alpha = 0.25$). The lasso models perform reasonably well on this data set, but they are still outperformed by the sparse group lasso models.

2.3. A simulation study

In this section we investigate the characteristics of the sparse group lasso estimator on simulated data sets. We are primarily interested in trends in the generalization error as α is varied and $\hat{\lambda}$ is selected by cross validation on a relatively small training set. We suspect that this trend will depend on the distribution of the data. We restrict our attention to multiclass data where the distribution of the features given the class is Gaussian. Loosely speaking, we suspect that if the differences in the data distributions are very sparse, i.e. the centers of the Gaussian distributions are mostly identical across classes, the lasso will produce models with the lowest generalization error. If the

data distribution is sparse, but not very sparse, then the optimal α is in the interval $(0, 1)$. For a dense distribution, with center differences between all or most classes, we expect the group lasso to perform best. The simulation study confirms this.

The mathematical formulation is as follows. Let

$$\mu = (\mu_1 \dots \mu_K)$$

where $\mu_i \in \mathbb{R}^p$ for $i = 1, \dots, K$ and $p = p_a + p_b$. Denote by $\mathcal{D}_\mu$ a data set consisting of N samples for each of the K classes – each sampled from the Gaussian distribution with centers $\mu_1, \dots, \mu_K$, respectively, and with a common covariance matrix Σ. Let $\hat{\lambda}$ be the smallest λ-value with the minimal estimated expected generalization error, as determined by cross validation on $\mathcal{D}_\mu$. Denote by $\mathrm{Err}_\mu(\lambda, \alpha)$ the generalization error of the model $\hat{\beta}(\lambda, \alpha)$ that has been estimated from the training set $\mathcal{D}_\mu$, by the sparse group lasso, for the given values of λ and α. Then let

$$Z_\mu(\alpha) = \mathrm{Err}_\mu(\hat{\lambda}, \alpha) - \mathrm{Err}_{\mathrm{Bayes}}(\mu)$$

where $\mathrm{Err}_{\mathrm{Bayes}}(\mu)$ is the Bayes rate. We are interested in trends in Z_μ, as a function of α, for different configurations of $\mu_1, \dots, \mu_K$. To be specific, we will sample $\mu_1, \dots, \mu_K$ from one of the following distributions:

- A *sparse* model distribution, where the first p_a entries of μ_i are i.i.d. with a distribution that is a mixture of the uniform distribution on $[-2, 2]$ and the degenerate distribution at 0 with point probability p_0.
- A *dense* model distribution, where the first p_a entries of μ_i are i.i.d. Laplace distributed with location 0 and scale b.

The last p_b entries are zero. We take $p_a = \lfloor 5/(1 - p_0) \rfloor$ throughout for the sparse model distribution. The within class covariance matrix Σ is constructed using features from the cancer site data set. Let Σ_0 be the empirical covariance matrix of p randomly chosen features. To avoid that the covariance matrix become singular we take

$$\Sigma = (1 - \delta)\Sigma_0 + \delta \mathbf{I}$$

for $\delta \in (0, 1)$.

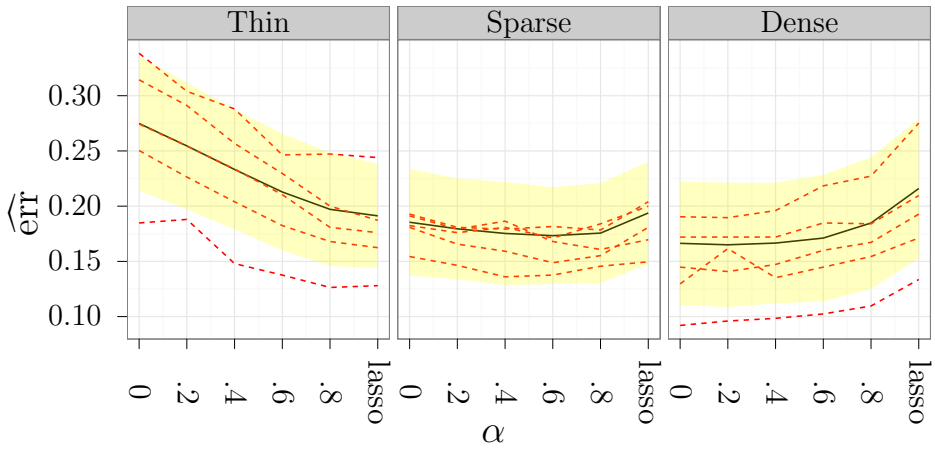


Figure 4: The estimated expected error gap (solid black line) for the three configurations. The central 95% of the distribution of $Z_\mu(\alpha)$ is shown as the shaded area on the plot. The error gap for 5 randomly selected μ-configurations is shown (red dashed lines).

The primary quantity of interest is

$$\mathrm{err}(\alpha) \stackrel{\mathrm{def}}{=} \mathrm{E}\left(Z_\mu(\alpha)\right), \tag{5}$$

the expectation being over μ and the data set $\mathcal{D}_\mu$. We are also interested in how well we can estimate the non-zero patterns of the μ_i's. Consider this as Kp two class classification problems, one for each parameter, where we predict the μ_{ij} to be non-zero if $\hat{\beta}_{ij}$ is non-zero, and μ_{ij} to be zero otherwise. We calculate the number of false positives, true positives, false negatives and true negatives. The positive predictive value (ppv) and the true positive rate (tpr) are of particular interest. The true positive rate measures how sensitive a given method is at discovering non-zero entries. The positive predictive value measures the precision with which the method is selecting the non-zero entries. We consider the following two quantities

$$\mathrm{tpr}(\alpha) \stackrel{\mathrm{def}}{=} \mathrm{E}\left[\mathrm{tpr}\left(\hat{\beta}(\hat{\lambda},\alpha)\right)\right] \text{ and } \mathrm{ppv}(\alpha) \stackrel{\mathrm{def}}{=} \mathrm{E}\left[\mathrm{ppv}\left(\hat{\beta}(\hat{\lambda},\alpha)\right)\right]. \tag{6}$$

In order to estimate the quantities (5) and (6) we sample M configurations of μ from one of the above distributions. For each configuration we sample a training and a test data set of sizes NK and $100K$, respectively. Using

the training data set we fit the model $\hat{\beta}(\hat{\lambda}, \alpha)$ and estimate $Z_\mu(\alpha)$ using the test data set. Estimates $\widehat{\text{err}}(\alpha)$, $\widehat{\text{tpr}}(\alpha)$ and $\widehat{\text{ppv}}(\alpha)$ are the corresponding averages over the M configurations.

For this study we chose $M = 100$, $N = 15$, $K = 25$, $p_b = 50$, $\delta = 0.25$ and the following three configuration distributions:

- *Thin* configurations, where the centers are distributed according to the sparse model distribution with $p_0 = 0.95$, as defined above.
- *Sparse* configurations, where the centers are distributed according to the sparse model distribution with $p_0 = 0.80$.
- *Dense* configurations, where the centers are distributed according to the dense model distribution with scale $b = 0.2$ and $p_a = 25$.

In Figure 4 we see that for thin configurations the lasso has the lowest estimated error gap, along with the sparse group lasso with $\alpha = 0.8$. For the sparse configurations the results indicate that the optimal choice of α is in the open interval $(0, 1)$, but in this case all choices of α result in a comparable error gap. For the dense configurations the group lasso is among the methods with the lowest error gap.

In Figure 5 we plotted the true positive rate for the three configurations. Except for the thin configurations, the lasso is markedly less sensitive than the sparse group and group lasso methods. However, looking at Figure 6 we see that the sparse group and group lasso methods have a lower precision than the lasso, except for the dense configurations. We note that the group lasso has the worst precision, except for the dense configurations.

3. The sparse group lasso algorithm

In this section we present the sparse group lasso algorithm. The algorithm is applicable to a broad class of loss functions. Specifically, we require that the loss function $f : \mathbb{R}^n \to \mathbb{R}$ is convex, twice continuously differentiable and bounded below. Additionally, we require that all quadratic approximations around a point in the sublevel set

$$\{\beta \in \mathbb{R}^n \mid f(\beta) + \lambda\Phi(\beta) \leq f(\beta_0) + \lambda\Phi(\beta_0)\}$$

are bounded below, where $\beta_0 \in \mathbb{R}^n$ is the initial point. The last requirement will ensure that all subproblems are well defined.

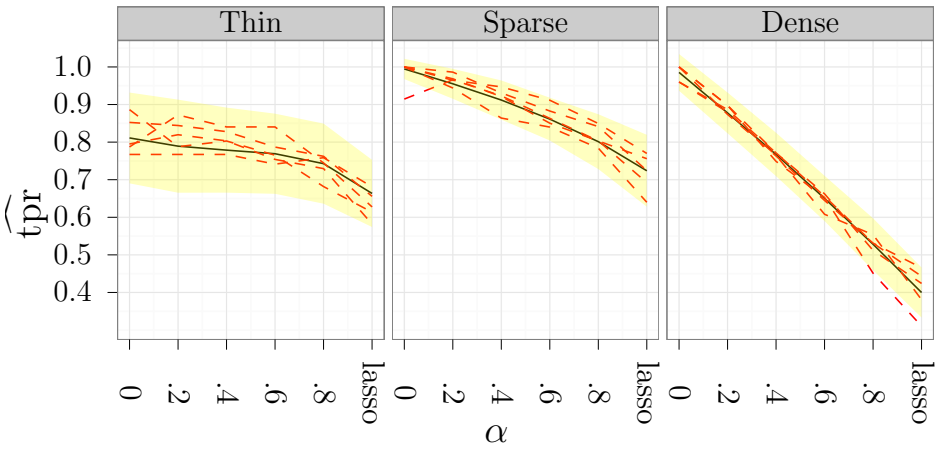


Figure 5: The estimated expected true positive rate (solid black line) for the three configurations. The central 95% of the distribution of tpr is shown as the shaded area on the plot. The true positive rate for 5 randomly selected μ-configurations is shown (red dashed lines).

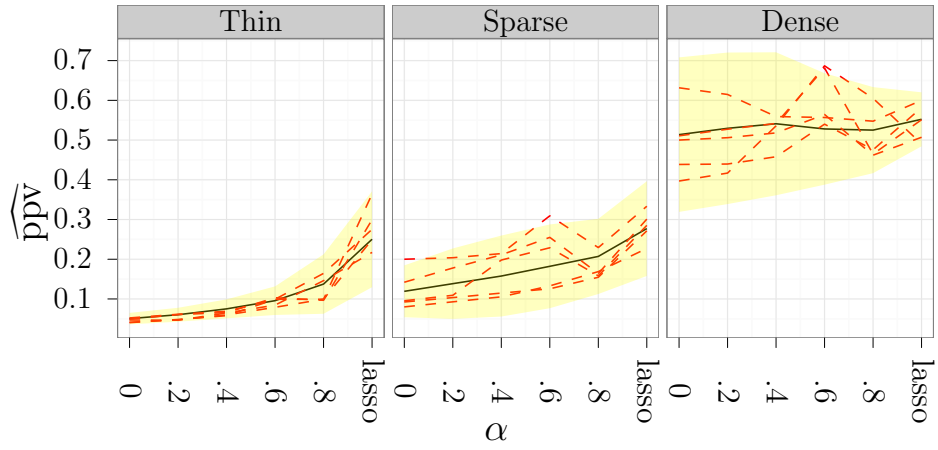


Figure 6: The estimated expected positive predictive value (solid black line) for the three configurations. The central 95% of the distribution of ppv is shown as the shaded area on the plot. The positive predictive value for 5 randomly selected μ-configurations is shown (red dashed lines).

The algorithm solves (1) for a decreasing sequence of λ values ranging from $\lambda_{\max}$ to a user specified $\lambda_{\min}$. The algorithm consists of four nested main loops:

- A numerical continuation loop, decreasing λ.
- An outer coordinate gradient descent loop (Algorithm 1).
- A middle block coordinate descent loop (Algorithm 2).
- An inner modified coordinate descent loop (Algorithm 3).

In Sections 3.3 to 3.5 we discuss the outer, middle and inner loop, respectively. In Section 3.6 we develop a method allowing us to bypass computations of large parts of the Hessian, hereby improving the performance of the middle loop. Section 4 provides a discussion of the available software solutions, as well as run-time performance of the current implementation.

Algorithms for solving the group lasso optimization problem have been around for some time, see, for example, [14] for an interesting application to multi-response linear regression. The sparse group lasso optimization problem is, however, more complicated, and group lasso algorithms cannot be used to compute a solution to the sparse group lasso optimization problem. Coordinate descent methods still constitute the core of our algorithm, and we give a short review tailored to this paper in Appendix A. See also [5, 6] for further details.

3.1. The sparse group lasso penalty

In this section we derive fundamental results regarding the sparse group lasso penalty.

We first observe that Φ is separable in the sense that if, for any group $J \in 1, \ldots, m$, we define the convex penalty $\Phi^{(J)} : \mathbb{R}^{n_J} \to \mathbb{R}$ by

$$\Phi^{(J)}(\hat{x}) \stackrel{\text{def}}{=} (1-\alpha)\gamma_J \left\|\hat{x}\right\|_2 + \alpha \sum_{i=1}^{n_J} \xi_i^{(J)} \left|\hat{x}_i\right|$$

then $\Phi(\beta) = \sum_{J=1}^{m} \Phi^{(J)}(\beta^{(J)})$. Separability of the penalty is required to ensure convergence of coordinate descent methods, see [5, 6], and see also Appendix A.

In a block coordinate descent scheme the primary minimization problem is solved by minimizing each block, one at a time, until convergence. We consider conditions ensuring that

$$0 \in \operatorname*{arg\,min}_{x \in \mathbb{R}^{n_J}} g(x) + \lambda \Phi^{(J)}(x) \tag{7}$$

for a given convex and twice continuously differentiable function $g : \mathbb{R}^{n_J} \to \mathbb{R}$. For $J = 1, \ldots, m$ a straightforward calculation shows that the subdifferential of $\Phi^{(J)}$ at zero is

$$\partial \Phi^{(J)}(0) = (1 - \alpha)\gamma_J B^{n_J} + \alpha \operatorname{diag}(\xi^{(J)}) T^{n_J}$$

where $B^n \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \mid \|x\|_2 \leq 1\}$, $T^n \stackrel{\text{def}}{=} [-1, 1]^n$ and where for $x \in \mathbb{R}^n$ $\operatorname{diag}(x)$ denotes the $n \times n$ diagonal matrix with diagonal x. For an introduction to the theory of subdifferentials see Chapter 4 in [15].

Proposition 1 below gives a necessary and sufficient condition for (7) to hold. Before we state the proposition the following definition is needed.

Definition 2. *For $n \in \mathbb{N}$ we define the map $\kappa : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ by*

$$\kappa(v, z)_i \stackrel{def}{=} \begin{cases} 0 & |z_i| \leq v_i \\ z_i - \operatorname{sgn}(z_i) v_i & \text{otherwise} \end{cases} \quad \text{for } i = 1, \ldots, n$$

and the function $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ by

$$K(v, z) \stackrel{def}{=} \|\kappa(v, z)\|_2^2 = \sum_{\{i \,|\, |z_i| > v_i\}} \left(z_i - \operatorname{sgn}(z_i) v_i\right)^2.$$

Proposition 1. *Assume given $a > 0$, $v, z \in \mathbb{R}^n$ and define the closed sets*

$$Y = z + \operatorname{diag}(v) T_n \quad \text{and} \quad X = aB^n + Y.$$

Then the following hold:

a. $\kappa(v, z) = \operatorname*{arg\,min}_{y \in Y} \|y\|_2$.

b. $0 \in X$ *if and only if* $K(v, z) \leq a^2$.

c. If $K(v, z) > a^2$ *then* $\operatorname*{arg\,min}_{x \in X} \|x\|_2 = \left(1 - a/\sqrt{K(v, z)}\right) \kappa(v, z)$.

The proof of Proposition 1 is given in Appendix C. Proposition 1 implies that (7) holds if and only if

$$\sqrt{K(\lambda\alpha\xi^{(J)}, \nabla g(0))} \leq \lambda(1-\alpha)\gamma_J.$$

The following observations will prove to be valuable. Note that we use $\preceq$ to denote coordinatewise ordering.

Lemma 1. *For any three vectors $v, z, z' \in \mathbb{R}^n$ the following hold:*

a. $K(v, z) = K(v, |z|)$.

b. $K(v, z) \leq K(v, z')$ *when* $|z| \preceq |z'|$.

Proof. (a) is a simple calculation and (b) is a consequence of the definition and (a). □

3.2. The λ-sequence

For sufficiently large λ values the only solution to (1) will be zero. We denote the infimum of these by $\lambda_{\max}$. By using the above observations it is clear that

$$\begin{aligned}
\lambda_{\max} &\stackrel{\text{def}}{=} \inf\left\{\lambda > 0 \;\middle|\; \hat{\beta}(\lambda) = 0\right\} \\
&= \inf\left\{\lambda > 0 \;\middle|\; \forall J = 1, \ldots, m : \sqrt{K(\lambda\alpha\xi^{(J)}, \nabla f(0)^{(J)})} \leq \lambda(1-\alpha)\gamma_J\right\} \\
&= \max_{J=1,\ldots,m} \inf\left\{\lambda > 0 \;\middle|\; \sqrt{K(\lambda\alpha\xi^{(J)}, \nabla f(0)^{(J)})} \leq \lambda(1-\alpha)\gamma_J\right\}.
\end{aligned}$$

It is possible to compute

$$\inf\left\{\lambda > 0 \;\middle|\; \sqrt{K(\lambda\alpha\xi^{(J)}, \nabla f(0)^{(J)})} \leq \lambda(1-\alpha)\gamma_J\right\}$$

by using the fact that the function $\lambda \to K(\lambda\alpha\xi^{(J)}, \nabla f(0)^{(J)})$ is piecewise quadratic and monotone.

3.3. Outer loop

In the outer loop a coordinate gradient descent scheme is used. In this paper we use the simplest form of this scheme. In this simple form the coordinate gradient descent method is similar to Newton's method; however the important difference is the way the non-differentiable penalty is handled. The convergence of the coordinate gradient descent method is not trivial and is established in [5].

The algorithm is based on a quadratic approximation of the loss function f, at the current estimate of the minimizer. The difference, Δ, between the minimizer of the penalized quadratic approximation and the current estimate is then a descent direction. A new estimate of the minimizer of the objective is found by applying a line search in the direction of Δ. We repeat this until a stopping condition is met, see Algorithm 1. Note that a line search is necessary in order to ensure global convergence. For most iterations, however, $t = 1$ will give sufficient decrease in the objective. With $q = \nabla f(\beta)$ and $H = \nabla^2 f(\beta)$ the quadratic approximation of f around the current estimate, β, is

$$\begin{aligned} q^T(x-\beta) + \frac{1}{2}(x-\beta)^T H(x-\beta) \\ = q^T x - q^T\beta + \frac{1}{2}x^T H x - \frac{1}{2}\left(\beta^T H x + x^T H\beta\right) + \frac{1}{2}\beta^T H\beta. \end{aligned}$$

H is symmetric, thus it follows that the quadratic approximation of f around β equals

$$Q(x) - q^T\beta + \frac{1}{2}\beta^T H\beta,$$

where $Q : \mathbb{R}^n \to \mathbb{R}$ is defined by

$$Q(x) \stackrel{\text{def}}{=} (q - H\beta)^T x + \frac{1}{2}x^T H x.$$

We have reduced problem (1) to the following penalized quadratic optimization problem

$$\min_{x\in\mathbb{R}^n} Q(x) + \lambda\Phi(x). \tag{8}$$

The convergence of Algorithm 1 is implied by Theorem 1e in [5]. This implies:

Proposition 2. *Every cluster point of the sequence $\{\beta_k\}_{k\in\mathbb{N}}$ generated by Algorithm 1 is a solution of problem (1).*

Algorithm 1 Outer loop. Solve (1) by coordinate gradient descent.

Require: $\beta = \beta_0$

repeat

Let $q = \nabla f(\beta)$, $H = \nabla^2 f(\beta)$ and $Q(x) = (q - H\beta)^T x + \frac{1}{2} x^T H x$.

Compute $\hat{\beta} = \arg\min_{x \in \mathbb{R}^n} Q(x) + \lambda \Phi(x)$.

Compute step size t and set $\beta = \beta + t\Delta$, for $\Delta = \beta - \hat{\beta}$.

until stopping condition is met.

Remark 1. *The convergence of Algorithm 1 is ensured even if H is a (symmetric) positive definite matrix approximating $\nabla^2 f(\beta)$. For high dimensional problems it might be computationally beneficial to take H to be diagonal, e.g. as the diagonal of $\nabla^2 f(\beta)$.*

3.4. Middle loop

In the middle loop the penalized quadratic optimization problem (8) is solved. The penalty Φ is block separable, i.e.

$$Q(x) + \lambda \Phi(x) = Q(x) + \lambda \sum_{J=0}^{p} \Phi^{(J)}(x^{(J)})$$

with $\Phi^{(J)}$ convex, and we can therefore use the block coordinate descent method over the blocks $x^{(1)}, \ldots, x^{(m)}$. The block coordinate descent method will converge to a minimizer even for non-differentiable objectives if the non-differentiable parts are block separable, see [6]. Since Φ is separable and Q is convex, twice continuously differentiable and bounded below, the block coordinate descent scheme converges to the minimizer of problem (8). Hence, our problem is reduced to the following collection of problems, one for each $J = 1, \ldots, m$,

$$\min_{\hat{x} \in \mathbb{R}^{n_J}} Q^{(J)}(\hat{x}) + \lambda \Phi^{(J)}(\hat{x}) \tag{9}$$

where $Q^{(J)} : \mathbb{R}^{n_J} \to \mathbb{R}$ is the quadratic function

$$\hat{x} \to Q(x^{(1)}, \ldots, x^{(J-1)}, \hat{x}, x^{(J+1)}, \ldots, x^{(m)})$$

up to an additive constant. We decompose an $n \times n$ matrix H into block matrices in the following way

$$H = \begin{pmatrix} H_{11} & H_{12} & \cdots & H_{1m} \\ H_{21} & H_{22} & \cdots & H_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ H_{m1} & H_{m2} & \cdots & H_{mm} \end{pmatrix}$$

where H_{IJ} is an $n_I \times n_J$ matrix. By the symmetry of H it follows that

$$\begin{aligned} Q^{(J)}(\hat{x}) &= \hat{x}^T (q - H\beta)^{(J)} + \frac{1}{2}\left(2\sum_I \hat{x}^T H_{JI} x^{(I)} - \hat{x}^T H_{JJ} x^{(J)} + \hat{x}^T H_{JJ} \hat{x}\right) \\ &= \hat{x}^T \left(q^{(J)} + [H(x-\beta)]^{(J)} - H_{JJ} x^{(J)}\right) + \frac{1}{2}\hat{x}^T H_{JJ} \hat{x} \end{aligned}$$

up to an additive constant. We may, therefore, redefine

$$Q^{(J)}(\hat{x}) \stackrel{\text{def}}{=} \hat{x}^T g^{(J)} + \frac{1}{2}\hat{x}^T H_{JJ} \hat{x}$$

where the *block gradient* $g^{(J)}$ is defined by

$$g^{(J)} \stackrel{\text{def}}{=} q^{(J)} + [H(x-\beta)]^{(J)} - H_{JJ} x^{(J)}. \tag{10}$$

For the collection of problems given by (9) a considerable fraction of the minimizers will be zero in practice. By Lemma 1 this is the case if and only if

$$\sqrt{K(\lambda\alpha\xi^{(J)}, g^{(J)})} \leq \lambda(1-\alpha)\gamma_J.$$

These considerations lead us to Algorithm 2.

3.5. Inner loop

Finally we need to determine the minimizer of (9), i.e. the minimizer of

$$\underbrace{Q^{(J)}(\hat{x}) + \lambda(1-\alpha)\gamma_J \left\|\hat{x}\right\|_2}_{\text{loss}} + \underbrace{\alpha \sum_{i=0}^{n_J} \xi_i^{(J)} \left|\hat{x}_i\right|}_{\text{penalty}}. \tag{11}$$

The two first terms of (11) are considered the loss function and the last term is the penalty. Note that the loss is not differentiable at zero (due to

Algorithm 2 Middle loop. Solve (8) by block coordinate descent.

repeat
  Choose next block index J according to the cyclic rule.
  Compute the block gradient $g^{(J)}$.
  if $\sqrt{K(\lambda\alpha\xi^{(J)}, g^{(J)})} \leq \lambda(1-\alpha)\gamma_J$ **then**
    Let $x^{(J)} = 0$.
  else
    Let $x^{(J)} = \underset{\hat{x}\in\mathbb{R}^{n_J}}{\arg\min}\, Q^{(J)}(\hat{x}) + \lambda\Phi^{(J)}(\hat{x})$.
  end if
until stopping condition is met.

the L_2-norm), thus we cannot completely separate out the non-differentiable parts. This implies that ordinary block coordinate descent is not guaranteed to converge to a minimizer. Algorithm 3 adjusts for this problem, and we have the following proposition.

Proposition 3. *For any $\epsilon > 0$ the cluster points of the sequence $\{\hat{x}_k\}_{k\in\mathbb{N}}$ generated by Algorithm 3 are minimizers of (11).*

Proof. Since $Q^{(J)}(0) + \lambda\Phi^{(J)}(0) = 0$ Algorithm 3 is a modified block coordinate descent scheme. Furthermore J is chosen such that (11) is not optimal at 0. We can therefore apply Lemma 4 in Appendix B, from which the claim follows directly. □

Hence, for a given block $J = 1, \ldots, m$ we need to solve the following two problems:

I. For each $j = 1, \ldots n_J$, compute a minimizer for the function

$$\mathbb{R} \ni \hat{x} \to Q^{(J)}(x^{(J)}, \ldots, x^{(J)}_{j-1}, \hat{x}, x^{(J)}_{j+1}, \ldots, x^{(J)}_{n_J}) + \lambda\Phi^{(J)}(x^{(J)}, \ldots, x^{(J)}_{j-1}, \hat{x}, x^{(J)}_{j+1}, \ldots, x^{(J)}_{n_J}).$$

II. Compute a descent direction at zero for (11).

Regarding I. Writing out the equation we see that in the j'th iteration we need to find the minimizer of the function $\omega : \mathbb{R} \to \mathbb{R}$ given by

$$\omega(\hat{x}) \stackrel{\text{def}}{=} c\hat{x} + \frac{1}{2}h\hat{x}^2 + \gamma\sqrt{\hat{x}^2 + r} + \xi\,|\hat{x}| \tag{12}$$

with $c = g_j^{(J)} + \sum_{i \neq j} (H_{JJ})_{ji} x_i$, $\gamma = \lambda(1-\alpha)\gamma_J$, $\xi = \lambda\alpha\xi_j^{(J)}$, $r = \sum_{i \neq j} x_i^2$, and where h is the j'th diagonal of the Hessian block H_{JJ}.

By convexity of f we conclude that $h \geq 0$. Lemma 2 below deals with the case $h > 0$. Since the quadratic approximation Q is bounded below the case $h = 0$ implies that $c = 0$, hence for $h = 0$ we have $\hat{x} = 0$.

Lemma 2. *If $h > 0$ then the minimizer $\hat{x}$ of ω is given as follows:*

a. If $r = 0$ or $\gamma = 0$ then

$$\hat{x} = \begin{cases} \frac{\xi+\gamma-c}{h} & \text{if } c > \xi + \gamma \\ 0 & \text{if } |c| \leq \xi + \gamma \\ \frac{-\xi-\gamma-c}{h} & \text{if } c < -\xi - \gamma \end{cases}$$

b. If $r > 0, \gamma > 0$ then $\hat{x} = 0$ if $|c| \leq \xi$ and otherwise the solution to

$$c + \operatorname{sgn}(\xi - c)\xi + h\hat{x} + \gamma \frac{\hat{x}}{\sqrt{\hat{x}^2 + r}} = 0.$$

Proof. Simple calculations will show the results. □

For case (b) in the above lemma we solve the equation by applying a standard root finding method.

Regarding II. For a convex function $f : \mathbb{R}^n \to \mathbb{R}$ and a point $x \in \mathbb{R}^n$, the vector

$$\Delta = -\operatorname*{arg\,min}_{\hat{x} \in \partial f(x)} \|\hat{x}\|_2$$

is a descent direction at x provided f is not optimal at x, see [15], Section 8.4. We may use this fact to compute a descent direction at zero for the function (11). By Proposition 1 it follows that $\Delta \in \mathbb{R}^n$ defined by

$$\Delta_i \stackrel{\text{def}}{=} -\begin{cases} 0 & \left|g_i^{(J)}\right| \leq \lambda\alpha\xi_i^{(J)} \\ g_i^{(J)} - \lambda\alpha\xi_i^{(J)} \operatorname{sgn}(g_i^{(J)}) & \text{otherwise} \end{cases}$$

is a descent direction at zero for the function (11).

Algorithm 3 Inner loop. Compute the minimizer of (11) by a modified coordinate descent scheme.

repeat

Choose next parameter index j according to the cyclic rule.

Compute

$$x_j^{(J)} = \arg\min_{\hat{x}\in\mathbb{R}} Q^{(J)}(x_1^{(J)}, \ldots, x_{j-1}^{(J)}, \hat{x}, x_{j+1}^{(J)}, \ldots, x_{n_J}^{(J)}) + \lambda\Phi^{(J)}(x^{(J)}, \ldots, x_{j-1}^{(J)}, \hat{x}, x_{j+1}^{(J)}, \ldots, x_{n_J}^{(J)})$$

if $\left\|x^{(J)}\right\|_2 < \epsilon$ and $Q^{(J)}(x^{(J)}) + \lambda\Phi^{(J)}(x^{(J)}) \geq 0$ **then**

Compute a descent direction, Δ, at zero for (11).

Use line search to find t such that $Q^{(J)}(t\Delta) + \lambda\Phi^{(J)}(t\Delta) < 0$.

Let $x^{(J)} = t\Delta$

end if

until stopping condition is met.

3.6. Hessian upper bound optimization

In this section we present a way of reducing the number of blocks for which the block gradient needs to be computed. The aim is to reduce the computational costs of the algorithm.

In the middle loop, Algorithm 2, the block gradient (10) is computed for all m blocks. We shall demonstrate that it is not necessary to compute the block gradient in order to determine if a block is zero, but that an upper bound of the block gradient is sufficient. Since the gradient, q, is already computed we focus on the term involving the Hessian. That is, for $J = 1, \ldots, m$, we compute a $b_J \in \mathbb{R}$ such that

$$\left|[H(x-\beta)]^{(J)}\right| \preceq b_J D_{n_J}$$

where $D_n \stackrel{\text{def}}{=} (1, 1, \ldots, 1) \in \mathbb{R}^n$. We define

$$t_J \stackrel{\text{def}}{=} \sup\left\{x \geq 0 \;\middle|\; \sqrt{K_J(\lambda\alpha\xi^{(J)}, |q^{(J)}| + xD_{n_J})} \leq \lambda(1-\alpha)\gamma_J\right\}$$

when $\sqrt{K_J(\lambda\alpha\xi^{(J)}, |q^{(J)}|)} \leq \lambda(1-\alpha)\gamma_J$ and otherwise let $t_J = 0$. When

$b_J < t_J$ it follows by Lemma 1 that

$$\begin{aligned} K_J(\lambda\alpha\xi^{(J)}, g^{(J)}) &= K_J(\lambda\alpha\xi^{(J)}, \left|g^{(J)}\right|) \\ &\leq K_J(\lambda\alpha\xi^{(J)}, \left|q^{(J)}\right| + b_J D_{n_J}) \\ &\leq \lambda^2(1-\alpha)^2\gamma_J^2 \end{aligned}$$

and by Proposition 1 this implies that the block J is zero. The above considerations lead us to Algorithm 4. Note that it is possible to compute the t_J's by using the fact that function

$$\mathbb{R} \ni x \to K_J(\lambda\alpha\xi^{(J)}, \left|q^{(J)}\right| + xD_{n_J})$$

is monotone and piecewise quadratic.

Algorithm 4 Middle loop with Hessian bound optimization.

repeat
  Choose next block index J according to the cyclic rule.
  Compute upper bound b_J.
  if $b_J < t_j$ **then**
    Let $x^{(J)} = 0$.
  else
    Compute $g^{(J)}$ and compute new $x^{(J)}$ (see Algorithm 2).
  end if
until stopping condition is met.

In Algorithm 4 it is unnecessary to compute the block gradient for all blocks, but only for those where $x^{(J)} \neq 0$ or when $b_j < t_J$. This will only be beneficial if we can efficiently compute a sufficiently good bound b_J. For a broad class of loss functions this can be done using the Cauchy-Schwarz inequality.

To assess the performance of the Hessian bound scheme we used our multinomial sparse group lasso implementation with and without bound optimization (and with $\alpha = 0.5$). Table 2 lists the ratio of the run-time without using bound optimization to the run-time with bound optimization, on the three different data sets. The Hessian bound scheme decreases the run-time for the multinomial loss function, and the ratio increases with the number of blocks m in the data set. The same trend can be seen for other values of α.

Data set	n	m	Ratio
Cancer	3.9k	217	1.14
Amazon	500k	10k	1.76
Muscle	220k	22k	2.47

Table 2: Timing the Hessian bound optimization scheme.

4. Software

We provide two software solutions in relation to the current paper. An R package, `msgl`, with a relatively simple interface to our multinomial and logistic sparse group lasso regression routines. In addition, a `C++` template library, `sgl`, is provided. The `sgl` template library gives access to the generic sparse group lasso routines. The R package relies on this library. The `sgl` template library relies on several external libraries. We use the Armadillo `C++` library [16] as our primary linear algebra engine. Armadillo is a `C++` template library using expression template techniques to optimize the performance of matrix expressions, see [17]. Furthermore we utilize several Boost libraries [18]. Boost is a collection of free peer-reviewed `C++` libraries, many of which are template libraries. For an introduction to these libraries see for example [19]. Use of multiple processors for cross validation and subsampling is supported through OpenMP [20].

The `msgl` R package is available from `CRAN`. The `sgl` library is available upon request.

4.1. Run-time performance

Table 3 lists run-times of the current multinomial sparse group lasso implementation for three real data examples. For comparison, the `glmnet` uses 5.2s, 8.3s and 137.0s, respectively, to fit the lasso path for the three data sets in Table 3. The `glmnet` is a fast implementation of the coordinate descent algorithm for fitting generalized linear models with the lasso penalty or the elastic net penalty [7]. Recently, support for multinomial group lasso has been added to `glmnet`, see [21]. However, `glmnet` cannot be used to fit models with the sparse group lasso penalty.

Data set	n	m	Lasso	Sparse group lasso		Group lasso
				$\alpha = 0.75$	$\alpha = 0.25$	
Cancer	3.9k	217	5.9s	4.8s	6.3s	6.0s
Muscle	220k	22k	25.0s	25.8s	37.7s	36.7s
Amazon	500k	10k	331.6s	246.7s	480.4s	285.1s

Table 3: Times for computing the multinomial sparse group lasso regression solutions for a lambda sequence of length 100, on a 2.20 GHz Intel Core i7 processor (using one thread). In all cases the sequence runs from $\lambda_{\max}$ to 0.002. The number of samples in the data sets Cancer, Muscle and Amazon are respectively 162, 107 and 1500. See also Table 1 and the discussions in Sections 2.2.1, 2.2.3 and 2.2.2 respectively.

5. Conclusion

We developed an algorithm for solving the sparse group lasso optimization problem with a general convex loss function. Furthermore, convergence of the algorithm was established in a general framework. This framework includes the sparse group lasso penalized negative-log-likelihood for the multinomial model, which is of primary interest for multiclass classification problems.

We implemented the algorithm as a `C++` template library. An R package is available for the multinomial and the logistic regression loss functions. We presented applications to multiclass classification problems using three real data examples. The multinomial group lasso solution achieved optimal performance in all three examples in terms of estimated expected misclassification error. In one example some sparse group lasso solutions achieved comparable performance based on fewer features. If there is a cost associated with the acquisition of each feature, this could be beneficial if we want to minimize the cost while optimizing the classification performance. In general, the sparse group lasso solutions provide more sparse solutions than the group lasso. Sparsity is generally of interest for model selection purposes and for interpretation of the model.

Appendix A. Block coordinate descent methods

In this section we review the theoretical basis of the optimization methods that we apply in the sparse group lasso algorithm. We use three slightly different methods: a coordinate gradient descent, a block coordinate descent and a modified block coordinate descent.

We are interested in unconstrained optimization problems on $\mathbb{R}^n$ where the coordinates are naturally divided into $m \in \mathbb{N}$ blocks with dimensions $n_i \in \mathbb{N}$ for $i = 1, \ldots, m$. We decompose the search space

$$\mathbb{R}^n = \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_m}$$

and denote by P_i the orthogonal projection onto the i'th block. For a vector $x \in \mathbb{R}^n$ we write $x = (x^{(1)}, \ldots, x^{(m)})$ where $x^{(1)} \in \mathbb{R}^{n_1}, \ldots, x^{(m)} \in \mathbb{R}^{n_m}$. For $i = 1, \ldots, m$ we call $x^{(i)}$ the i'th *block* of x. We assume that the objective function $F : \mathbb{R}^n \to \mathbb{R}$ is bounded below and of the form

$$F(x) = f(x) + \sum_{i=1}^{m} h_i(x^{(i)})$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is convex and each $h_i : \mathbb{R}^{n_i} \to \mathbb{R}$, for $i = 1, \ldots, m$ are convex. Furthermore, we assume that for any $i = 1, \ldots, m$ and any $x_0 = (x_0{}^{(1)}, \ldots, x_0{}^{(m)})$ the function

$$\mathbb{R}^{n_i} \ni \hat{x} \to F(x_0{}^{(1)}, \ldots, x_0{}^{(i-1)}, \hat{x}, x_0{}^{(i+1)}, \ldots, x_0{}^{(m)})$$

is hemivariate. A function is said to be *hemivariate* if it is not constant on any line segment of its domain.

Appendix A.1. Coordinate gradient descent

Algorithm 5 Coordinate gradient descent scheme.

repeat

Compute quadratic approximation Q of f around the current point x.
Compute search direction

$$x^{new} = \arg\min_{\hat{x} \in \mathbb{R}^n} Q(\hat{x}) + \sum_{i=1}^{m} h_i\left(\hat{x}^{(i)}\right).$$

Let $\Delta = x - x^{new}$ and compute step size t using the Armijo rule and let $x \leftarrow x + t\Delta$.

until stopping condition is met.

For this scheme we make the additional assumption that f is twice continuously differentiable everywhere. The scheme is outlined in Algorithm 5, where the step size is chosen by the Armijo rule outlined in Algorithm 6. Theorem 1e in [5] implies the following:

Algorithm 6 Armijo rule.

Require: $a \in (0, 0.5)$ and $b \in (0, 1)$
  Let $\delta = \nabla f(x)^T \Delta + \sum_{i=1}^{m} (h_i(x_i + \Delta_i) - h_i(x_i))$.
  while $F(x + t\Delta) > F(x) + ta\delta$ **do**
    $t \leftarrow bt$.
  end while

Corollary 1. *If f is twice continuously differentiable then every cluster point of the sequence $\{x_k\}_{k\in\mathbb{N}}$ generated by Algorithm 5 is a minimizer of F.*

Appendix A.2. Block coordinate descent

Algorithm 7 Block coordinate descent.

  repeat
    Choose next block index i according to the cyclic rule.
    $x^{(i)} \leftarrow \arg\min_{\hat{x}\in\mathbb{R}^{n_i}} F(\hat{x} \oplus P_i^{\perp} x)$.
  until some stopping condition is met.

The block coordinate descent scheme is outlined in Algorithm 7. By Corollary 2 below the block coordinate descent method converges to a *coordinatewise minimum*.

Definition 3. *A point $p \in \mathbb{R}^n$ is said to be a coordinatewise minimizer of F if for each block $i = 1, \ldots, m$ it holds that*

$$F(p + (0, \ldots, 0, d_i, 0, \ldots, 0)) \geq F(p) \text{ for all } d_i \in \mathbb{R}^{n_i}.$$

If f is differentiable then by Lemma 3 the block coordinate descent method converges to a minimizer. Lemma 3 below is a simple consequence of the separability of F.

Lemma 3. *Let $p \in \mathbb{R}^n$ be a coordinatewise minimizer of F. If f is differentiable at p then p is a stationary point of F.*

Proposition 5.1 in [6] implies the following:

Corollary 2. *For the sequence $\{x_k\}_{k\in\mathbb{N}}$ generated by the block coordinate descent algorithm (Algorithm 7) it holds that every cluster point of $\{x_k\}_{k\in\mathbb{N}}$ is a coordinatewise minimizer of F.*

Algorithm 8 Modified coordinate descent loop.

repeat
  Let $i \leftarrow i + 1 \mod m$.
  $x^{(i)} \leftarrow \arg\min_{\hat{x} \in \mathbb{R}^{n_i}} F(\hat{x} \oplus P_i^{\perp} x)$.
  if $\|x - p\|_2 < \epsilon$ and $F(x) \geq F(p)$ **then**
    Compute descent direction Δ at p for F.
    Use line search to find t such that $F(p + t\Delta) < F(p)$.
    Let $x^{(i)} \leftarrow p + t\Delta$.
  end if
until stopping condition is met.

Appendix B. Modified block coordinate descent

For this last scheme we make the additional assumption that f is twice continuously differentiable everywhere except at a given non-optimal point $p \in \mathbb{R}^n$. In this case the block coordinate descent method is no longer guaranteed to be globally convergent, as it may get stuck at p. One immediate solution to this is to compute a descent direction at p, then use a line search to find a starting point x_0 with $F(x_0) < F(p)$. Since f is differentiable on the sublevel set $\{x \in \mathbb{R}^n \mid F(x) < F(p)\}$ it follows by the results above that the cluster points of the generated sequence are stationary points of F. This procedure is not efficient since it discards a carefully chosen starting point. We apply the modified coordinate descent loop, outlined in Algorithm 8, instead.

Lemma 4. *Assume that f is differentiable everywhere except at $p \in \mathbb{R}^n$, and that F is not optimal at p. Then for any $\epsilon > 0$ the cluster points of the sequence $\{x^k\}_{k \in \mathbb{N}}$ generated by Algorithm 8 are minimizers of F.*

Proof. Let z be a cluster point of $\{x^k\}$. By Corollary 2, z is a coordinatewise minimizer of F. Then Lemma 3 implies that z is either p or a stationary point of F. We shall show by contradiction that p is not a cluster point of $\{x^k\}_{k \in \mathbb{N}}$, thus assume otherwise. The sequence $\{F(x^k)\}_{k \in \mathbb{N}}$ is decreasing; hence, if we can find a $k' \in \mathbb{N}$ such that $F(x^{k'}) < F(p)$ we reach a contradiction (since this would conflict with the continuity of F). Choose k' such that $\left\|x^{k'} - p\right\|_2 < \epsilon$. Since we may assume that $F(x^{k'}) \geq F(p)$ it follows by the definition of Algorithm 8 that $F(x^{k'+1}) < F(p)$. □

Appendix C. Proof of Proposition 1

(a) Straightforward.

(b) If $\|\kappa(v,z)\|_2 \leq a$ then $-\kappa(v,z) \in aB^n$ hence $0 \in X$. For the other implication simply choose $y_0 \in Y$ such that $-y_0 \in aB^n$ and note that $\|\kappa(v,z)\|_2 \leq \|y_0\|_2 \leq a$.

(c) Assume $\|\kappa(v,z)\|_2 > a$, and let $x^* = (1 - a/\|\kappa(v,z)\|_2)\kappa(v,z)$. Then $x^* \in X$ and $\|x^*\|_2 = \|\kappa(v,z)\|_2 - a$. The point x^* is in fact a minimizer. To see this let $x' \in X$, that is we have

$$x' = z + as + \mathrm{diag}(v)t$$

for some $s \in B^n$ and $t \in T_n$. It follows, by the triangle inequality and (a), that

$$\|x'\|_2 + a \geq \|x' - as\|_2 = \|z + \mathrm{diag}(v)t\|_2 \geq \|\kappa(v,z)\|_2 .$$

So $\|x'\|_2 \geq \|\kappa(v,z)\|_2 - a = \|x^*\|_2$ and since X is convex and $x \to \|x\|_2$ is strictly convex the found minimizer x^* is the unique minimizer.

[1] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society, Series B 58 (1994) 267–288.

[2] L. Meier, S. V. D. Geer, P. Bhlmann, E. T. H. Zrich, The group lasso for logistic regression, Journal of the Royal Statistical Society, Series B.

[3] J. Friedman, T. Hastie, R. Tibshirani, A note on the group lasso and a sparse group lasso, Tech. Rep. arXiv:1001.0736, comments: 8 pages, 3 figs (Jan 2010).

[4] N. Simon, J. Friedman, T. Hastie, R. Tibshirani, A sparse group lasso, Journal of Computational and Graphical Statistics.

[5] P. Tseng, S. Yun, A coordinate gradient descent method for nonsmooth separable minimization, Mathematical Programming 117 (1) (2009) 387–423.

[6] P. Tseng, Convergence of a block coordinate descent method for nondifferentiable minimization, Journal of optimization theory and applications 109 (3) (2001) 475–494.

[7] J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, Journal of Statistical Software 33 (1) (2010) 1–22.
URL http://www.jstatsoft.org/v33/i01/

[8] J. Lu, G. Getz, E. Miska, E. Alvarez-Saavedra, J. Lamb, D. Peck, A. Sweet-Cordero, B. Ebert, R. Mak, A. Ferrando, et al., Microrna expression profiles classify human cancers, nature 435 (7043) (2005) 834–838.

[9] S. Liu, Z. Liu, J. Sun, L. Liu, Application of synergetic neural network in online writeprint identification, International Journal of Digital Content Technology and its Applications 5 (3) (2011) 126–135.

[10] M. Bakay, Z. Wang, G. Melcon, L. Schiltz, J. Xuan, P. Zhao, V. Sartorelli, J. Seo, E. Pegoraro, C. Angelini, et al., Nuclear envelope dystrophies show a transcriptional fingerprint suggesting disruption of rb–myod pathways in muscle regeneration, Brain 129 (4) (2006) 996.

[11] Y. Kim, S. Kwon, S. Heun Song, Multiclass sparse logistic regression for classification of multiple cancer types using gene expression data, Comput. Stat. Data Anal. 51 (3) (2006) 1643–1655. doi:10.1016/j.csda.2006.06.007.
URL http://dx.doi.org/10.1016/j.csda.2006.06.007

[12] T. Hastie, R. Tibshirani, J. H. Friedman, The elements of statistical learning: data mining, inference, and prediction: with 200 full-color illustrations, New York: Springer-Verlag, 2001.

[13] A. Frank, A. Asuncion, UCI machine learning repository (2010).
URL http://archive.ics.uci.edu/ml

[14] T. Simil, J. T. I. Selection, S. In, T. Simil, J. Tikka, Input selection and shrinkage in multiresponse linear regression, Computational Statistics and Data Analysis 52 (2007) 406–422.

[15] D. Bertsekas, A. Nedić, A. Ozdaglar, Convex analysis and optimization, Athena Scientific optimization and computation series, Athena Scientific, 2003.

[16] C. Sanderson, Armadillo: An open source c++ linear algebra library for fast prototyping and computationally intensive experiments, Tech. rep., NICTA (October 2010).

[17] D. Eddelbuettel, C. Sanderson, Rcpparmadillo : accelerating r with high-performance c++ linear algebra, Computational Statistics & Data Analysis.

[18] Boost c++ libraries.
URL `http://www.boost.org`

[19] R. Demming, D. Duffy, Introduction to the Boost C++ libraries, Datasim Education, 2010.

[20] The openmp api specification for parallel programming.
URL `http://www.openmp.org`

[21] N. Simon, J. Friedman, T. Hastie, A blockwise descent algorithm for group-penalized multiresponse and multinomial regression.

Chapter 6

Article: Modeling tissue contamination

M. Vincent, K. Perell, F. C. Nielsen, G. Daugaard, and N. R. Hansen. Modeling tissue contamination to improve molecular identification. *Bioinformatics*

BIOINFORMATICS

Vol. 00 no. 00 2005
Pages 1–6

Modeling tissue contamination to improve molecular identification of the primary tumor site of metastases

Martin Vincent[1]*, Katharina Perell[2], Finn Cilius Nielsen[3], Gedske Daugaard[2] and Niels Richard Hansen[1]

[1]Department of Mathematical Sciences, University of Copenhagen, 2100 Copenhagen Ø, Denmark
[2]Department of Oncology and [3]Center for Genomic Medicine, Copenhagen University Hospital, Rigshospitalet, 2100 Copenhagen Ø, Denmark

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: Contamination of cancer tissue by surrounding benign (non-cancer) tissues is a concern for molecular cancer diagnostics. This is because an observed molecular signature will be distorted by the surrounding benign tissue, possibly leading to incorrect diagnosis. One example is molecular identification of the primary tumor site of metastases, since biopsies of metastases typically contain a significant amount of benign tissue.

Results: A model of tissue contamination is presented. This contamination model works independently of the training of a molecular predictor, and it can be combined with any predictor model. The usability of the model is illustrated on primary tumor site identification of liver biopsies. Specifically on a human data set consisting of microRNA expression measurements of primary tumor samples, benign liver samples and liver metastases. For this data set the best contamination model decreased the overall expected prediction error from 60% to 35%. Most notably, a decrease from 80% to 35% was seen for metastases with low tumor content.

Availability: http://www.math.ku.dk/∼richard/msgl/

Contact: vincent@math.ku.dk

1 INTRODUCTION

Several studies have considered molecular predictors for primary tumor site identification, see Ramaswamy *et al*., 2001, Lu *et al*., 2005 and Rosenfeld *et al*., 2008. These studies all report an error rate of around 10% in predicting the primary tumor site from primary tumor samples, which is consistent with our findings, see Section 3. However, the performance of molecular predictors on metastatic samples is less clear. Most studies assess the performance of their predictor using a combination of primary and metastatic samples, with an unbalanced metastatic sample set. Moreover, samples of metastatic cancers, which are difficult to diagnose by conventional diagnostic methods, are generally underrepresented in the validation. To correctly validate the performance of a predictor, the validation samples must be representative for the samples that the predictor is intended to be used on. For the majority of patients with metastatic cancer, identification of the primary tumor site relies on small formalin-fixed paraffin-embedded (FFPE) needle biopsies (core biopsies) from metastatic lesions. Hence, we argue that a molecular predictor designed to assist in the diagnosis of patients with metastatic cancer must be compatible with and validated on core biopsy samples.

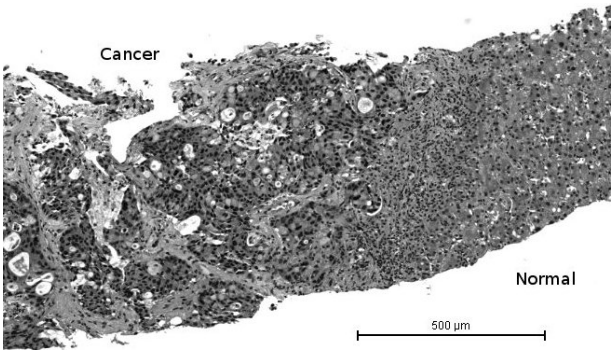


Fig. 1: Micrograph of a liver core biopsy with cancer. Cancer as well as normal (benign) tissue can clearly be seen.

For the development of such a predictor it seems preferable to train it exclusively on core biopsies of metastatic tissue. It is, however, difficult to obtain a sufficient amount of metastatic tumor biopsies of known primary tumor origin. Furthermore, a larger technical variation due the smaller amount of processed material, and the varying tumor content, make it difficult in practice to rely on core biopsies only. Therefore, previous studies as well as ours have relied on primary tumor samples in the training set.

These considerations leads to the central problem that we address in this paper; *when core biopsy samples are scarce or completely absent, how can we best adapt primary tumor samples to build a molecular predictor of metastatic tumor samples*?

Such an adaption is generally referred to as a *domain adaption*. The problem being that the distribution of the cases to be predicted (the target domain) is not the same as the distribution of the cases used for training (the source domain), see e.g. Mansour *et al*., 2009 and Daumé *et al*., 2006. To overcome this problem we explicitly

*to whom correspondence should be addressed

model how the target domain is related to the source domain and use this model to train a predictor. A central difference between the source and target domains in our setting is caused by the fact that a core biopsy from a metastatic lesion is contaminated by tissue surrounding the tumor cells, that is, benign tissue from the biopsy site unrelated to the tumor, see Figure 1. Another difference can arise if the molecular signature of the metastatic tumor deviates from the signature of the primary tumor, see, for example, Ramaswamy *et al*., 2002 or Albini *et al*., 2008 for a discussion of the biology of metastases. Our results, as well as other recent findings, Elloumi *et al*., 2011, suggest that tissue contamination may result in a decline of performance for molecular predictors. Contamination appears to be particularly problematic for identification of primary tumor sites based on microRNA expression. The problem may not be as severe for other molecular predictors, such as predictors based on messenger RNA. Our contamination model and suggested methodology are, however, not specific to microRNA expression or the technological platform used. It is a general, computational model, which is broadly applicable whenever tissue contamination constitutes a problem.

For primary tumor site identification of liver core biopsies we show that an unmodified predictor trained solely on microRNA expression measurements from primary tumors and benign liver samples has a high error rate (60% in our case), see Section 3.2. This error rate is reduced to around 35% using our contamination model and suggested domain adaption procedure. Finally, we show that metastatic tumor samples from the liver can be combined with our model in the training phase. This combined approach can, in our case, bring the error rate further down to around 30%.

2 METHODS

2.1 Domain adaption

To present our contamination model in an appropriate context we briefly review the domain adaption terminology. The goal is to predict samples drawn from a distribution $\mathcal{T}$ called the *target distribution* or *target domain*. However, none or only a small number of samples drawn from $\mathcal{T}$ are available for training. The philosophy of domain adaption is to use a related distribution $\mathcal{S}$, called the *source distribution* or *source domain*, to construct a target domain predictor. Typically, either the source distribution or a predictor trained on samples from the source distribution are adapted to the target domain – perhaps using (the few) available target samples.

In our setting the source distribution is the distribution of molecular signatures for primary tumor and benign liver samples (resections). The target distribution is the distribution of molecular signatures from liver core biopsies. The relation between the source distribution and the target distribution is made explicit in Section 2.2 below. We suggest to apply the following general domain adaption strategy to our problem.

1. Specify a domain adaption *model*, that is, specify a map
$$F : \mathcal{F}_{\text{source}} \to \mathcal{F}_{\text{target}}$$
where $\mathcal{F}_{\text{source}}$ and $\mathcal{F}_{\text{target}}$ denote the sets of relevant source and target distributions, respectively.
2. With $\widehat{\mathcal{S}}_n$ the empirical distribution of n source samples use the plug-in estimate $F(\widehat{\mathcal{S}}_n)$ as an estimate of the target distribution.
3. Generate an approximate target distribution $\mathcal{T}_{\text{sim}}$ by simulation of artificial target samples based on $F(\widehat{\mathcal{S}}_n)$.
4. Use $\mathcal{T}_{\text{sim}}$ for training of a predictor.

The domain adaption model F provides a special kind of relation between source and target – for any given source $\mathcal{S}$ there is only one related target, the *model target* $F(\mathcal{S})$. In practice, the model F will contain unknown components that have to specified or estimated from data as well.

Note that three approximations are involved, namely

$$\mathcal{T} \underbrace{\approx}_{\text{depends on the model}} F(\mathcal{S}) \underbrace{\approx}_{\text{small for } n \text{ large}} F(\widehat{\mathcal{S}}_n) \underbrace{\approx}_{\text{small}} \mathcal{T}_{\text{sim}}.$$

2.2 Contamination models

A simple model of the molecular signature from a liver contaminated core biopsy is

$$\alpha \times \text{primary tumor signature} + (1-\alpha) \times \text{normal liver signature}$$

where $\alpha \in [0,1]$ is the relative amount of tumor content. This is a plausible model on the molecular level but the contamination is not necessarily additive on the measured scale. We therefore need to transform the measurements using a suitable *scale function* $f : \mathbb{R}^p \to \mathbb{R}^p$, which is a function

$$f(x_1, \ldots, x_p) = (g(x_1), \ldots, g(x_p))$$

for a continuous strictly monotone function $g : \mathbb{R} \to \mathbb{R}$.

Letting $(X, Y) \in \mathbb{R}^p \times \{1, \ldots, K\}$ denote a pair of random variables with X representing the molecular signature for a primary tumor with site label Y, the distribution of (X, Y) is the source distribution. With $Z \in \mathbb{R}^p$ a random variable representing the contamination we introduce

$$U(y) \stackrel{\text{def}}{=} f\left(\alpha f^{-1}(X) + (1-\alpha) f^{-1}(Z)\right) \mid Y = y$$

for f a scale function and $\alpha \in [0,1]$ another random variable representing the relative amount of tumor content. The variables X, Z and α are assumed conditionally independent given Y as illustrated in Figure 2a. With $Y_{\mathcal{T}}$ the marginal distribution of class labels in the target distribution, the distribution of $(U(Y_{\mathcal{T}}), Y_{\mathcal{T}})$ constitutes our model target. The model is specified by choosing a scale function and conditional distributions of Z and α given Y.

We will consider domain adaption models for two particular scale functions. The *linear scale* function is given by simply taking f to be the identity, thus assuming that the contamination is additive on the measured scale. The data example used in this paper consists of measurements of microRNA (miRNA, a small non-coding RNA molecule) expression using quantitative PCR (qPCR), see, for example, Vaerman *et al*., 2004 or VanGuilder *et al*., 2008 for an introduction to qPCR. For qPCR a logarithmic scale function that models the relation between actual miRNA concentration and the measured quantity is appropriate. The *log scale* function is given by

$$g(\delta) \stackrel{\text{def}}{=} -K \log \delta,$$

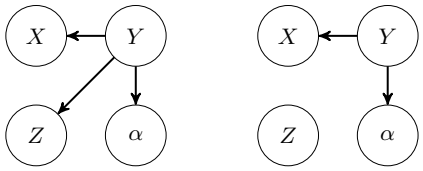
for $\delta \in (0,1]$ and a constant $K > 0$. The log scale function can be derived based on theoretical considerations for qPCR reactions. The constant $\eta = e^K - 1$ is the amplification efficiency, and we use a standard value of $\eta = 0.8$ throughout.

2.3 Simulation of artificial core biopsies

The approximate target distribution $\mathcal{T}_{\text{sim}}$ is generated as a weighted empirical distribution of a total of M simulated samples. First, the class labels $y_1, \ldots, y_M$ are sampled or chosen. Based on the source data set (the primary tumors) we sample $x_1, \ldots, x_M$ independently given $y_1, \ldots, y_M$ such that $x_i \mid y_i$ is sampled from the conditional empirical distribution of the source data given y_i. That is, x_i is drawn with replacement from the source data with class label y_i. Given a data set from the contamination distribution we draw M samples with replacement, $z_1, \ldots, z_M$, from the contamination data. In addition, we sample $\alpha_1, \ldots, \alpha_M$ independently given $y_1, \ldots, y_M$ such that $\alpha_i \mid y_i$ has the desired distribution. For a given scale function f we compute

$$u_i = f\left(\alpha_i f^{-1}(x_i) + (1-\alpha_i) f^{-1}(z_i)\right),$$

and the empirical distribution of the samples $(u_1, y_1), \ldots, (u_M, y_M)$ with weights $\omega_1, \ldots, \omega_M$ form the approximate target distribution $\mathcal{T}_{\text{sim}}$. The



(a) Minimal (b) Actual

Fig. 2: The random variables X, Y, Z and α represent the molecular signature from the primary tumor, its site label, the signature from liver contamination and the tumor proportion, respectively. A minimal assumption (a) for our domain adaption model is that X, Z and α are conditionally independent given Y. The actual assumption made (b) is that Z is marginally independent of the other variables and that X and α are conditionally independent given Y.

weights can be chosen to achieve a desired distribution of class labels in $\mathcal{T}_{\text{sim}}$, for example, to match the distribution of class labels for the primary tumor data set. The simulation assumes the conditional independence structure illustrated in Figure 2b. If the less restrictive conditional independence structure illustrated in Figure 2a is assumed, we need to adjust the simulation to draw z_i conditionally on y_i, but this requires knowledge of a class dependent contamination distribution.

2.4 Multinomial group lasso regression

The data set used in this paper consists of miRNA signatures from 9 different classes, see Table 1. For class prediction we use multinomial logistic regression, which is the multiclass extension of logistic regression. The predictor is trained using a group lasso penalized likelihood approach as treated in details in Vincent *et al*., 2012.

We briefly review the multinomial group lasso regression method. Consider a prediction problem with K classes, N samples, and p features. Assume given a data set $(x_1, y_1), \ldots, (x_N, y_N)$ where, for all $i = 1, \ldots, N$, $x_i \in \mathbb{R}^p$ is the observed feature vector and $y_i \in \{1, \ldots, K\}$ is the categorical class label. With $h : \{1, \ldots, K\} \times \mathbb{R}^p \to \mathbb{R}$ defined as

$$h(l, \eta) \stackrel{\text{def}}{=} \frac{\exp(\eta_l)}{\sum_{k=1}^{K} \exp(\eta_k)},$$

the (symmetric) multinomial model is given by

$$P(Y = l \mid x) = h(l, \beta^{(0)} + \beta x).$$

Here the parameters are organized as the K-dimensional vector $\beta^{(0)}$ of intercept parameters and the $K \times p$ matrix

$$\beta \stackrel{\text{def}}{=} \left(\beta^{(1)} \cdots \beta^{(p)}\right), \tag{1}$$

with $\beta^{(j)} \in \mathbb{R}^K$ the parameters associated with the j'th feature. The group lasso maximum likelihood estimator of $(\beta^{(0)}, \beta)$ is the minimizer of the group lasso penalized negative log-likelihood,

$$-\sum_{i=1}^{N} \omega_i \log h(y_i, \beta^{(0)} + \beta x_i) + \lambda \sum_{J=1}^{p} \left\| \beta^{(J)} \right\|_2, \tag{2}$$

where $\omega_1, \ldots, \omega_N$ are sample weights. Here $|| \cdot ||_2$ is the 2–norm on $\mathbb{R}^K$. The penalization results in feature selection meaning that for some features the corresponding parameter vector is estimated to be 0. The regularization parameter $\lambda > 0$ is a tuning parameter and the larger λ is the fewer features are selected.

Table 1. Number of samples included in the study. For the metastases the numbers a and b shown as (a/b) are the numbers of low and high tumor content samples, respectively.

Class description	Primaries (resections)	Metastases (core biopsies)
Breast cancer	17	7 (5/2)
Colorectal cancer	20	12 (8/4)
Gastric/Cardia cancer	18	12 (8/4)
Pancreatic cancer	20	10 (5/5)
Squamous cell cancers (of different origins)	16	12 (6/6)
Hepatocellular carcinoma	17	3
Cholangiocarcinoma	20	4
Cirrhotic liver	17	8
Normal liver	20	7

2.5 Biological samples

Samples were selected from 240 independent patients and consist of a total of 128 resected primary tumors of different origin (representing 7 primary tumor classes), 60 liver core biopsies from metastatic tumors of known origin (representing the same predefined 7 primary tumor classes), 37 resected and 15 core biopsies from benign liver. The 9 classes were further grouped into 3 groups; metastatic cancers, primary liver cancers and benign liver, see Table 1. Samples were obtained from The University Hospital of Copenhagen, Denmark. The sample set is a subset of the samples used in Perell *et al*., 2013. All samples were archived formalin-fixed paraffin embedded (FFPE) tissues (dated 2000-2012). All primary tumor resections were cut into one section of 10 µm and laser-dissected before being processed in order to remove the surrounding benign tissue. Core biopsies were cut into two sections of 5 µm according to standard pathological procedure – no micro-dissection was performed. Malignant core biopsy samples were required to have a minimum tumor content of 10%. Tumor content was defined as tumor- and stromal-cells. All samples were from independent patients, hence no patient overlap between primary tumor samples and metastatic samples were accepted. All samples were reviewed by an independent pathologist to confirm the reference diagnosis and to estimate the percentage of tumor cells. Based on the tumor cell estimate, the core biopsy samples were divided into two groups representing high (above 50%) and low (below 50%) tumor content. For each sample the expression levels of 377 miRNAs were measured using quantitative real time PCR (TaqManlow density array cards, human MicroRNA array A, Applied Biosystems) according to manufacturers instructions.

Before any multinomial predictor was trained the artificial core biopsy data were simulated and data were preprocessed. Simulation and preprocessing were executed in the following order.

1. Controls and miRNAs not expressed in the primary tumor samples were removed.
2. Linear and log scale artificial core biopsies were simulated, see Section 2.3.
3. All samples, that is, primary tumor samples, core biopsy samples and the artificial core biopsy samples, were normalized and then standardized as described below.

The data were first *normalized* by centering and scaling the individual samples to mean 0 and variance 1. The purpose of normalization is to

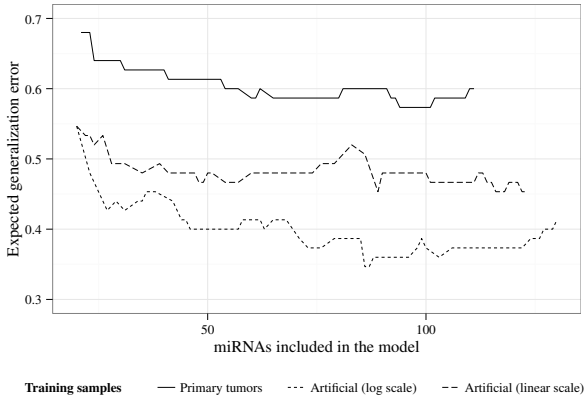


Fig. 3: Estimated expected generalization error for prediction of liver core biopsies. The error is shown as a function of miRNAs included in the predictor. The predictors were trained using primary tumor samples or artificial core biopsy samples derived from primary tumor samples using either a linear or a log scale function.

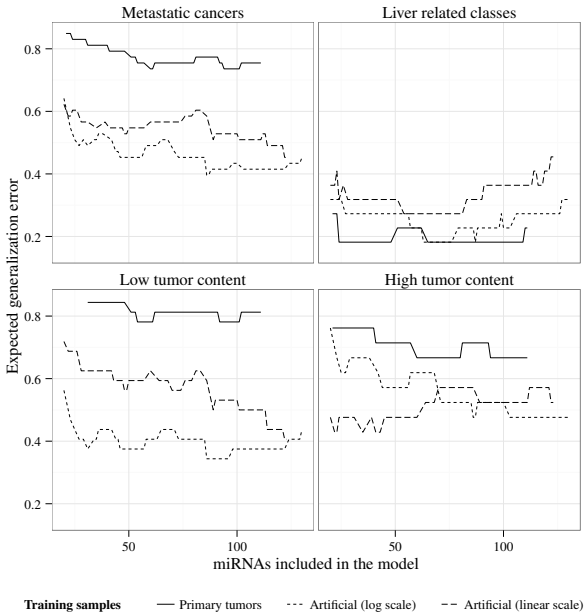


Fig. 4: Estimated expected generalization error for prediction of liver core biopsies. The error is shown as a function of miRNAs included in the predictor. The results are stratified according to the four following subgroups: metastatic cancers, liver related classes (benign liver, hepatocellular carcinoma and cholangiocarcinoma), metastatic cancer samples with low tumor content and metastatic cancer samples with high tumor content.

remove technical (non-biological) variation. The data were *standardized* by centering and scaling the expression measurements for each miRNA across the samples. This is to ensure that differences in scale will not influence the variable selection. The centers and scales were estimated using the primary tumor samples and applied for standardization of the primary tumor samples as well as the core biopsy and artificial core biopsy samples. That is, the standardized sample $\tilde{x} \in \mathbb{R}^p$ of a sample $x \in \mathbb{R}^p$ is given by

$$\tilde{x}_i = \frac{x_i - \hat{\mu}_i}{\hat{\sigma}_i} \qquad \text{for } i = 1, \ldots, p$$

where $\hat{\mu}_i$ and $\hat{\sigma}_i$ denote the empirical mean and standard deviation, respectively, for the i'th miRNA in the primary tumor data set. Note that the order in which normalization and standardization are applied matters.

3 RESULTS

In this section we present the results obtained by testing our domain adaption model on the miRNA expression data set described in Section 2.5. The domain adaption model was used with a linear and a log scale function, as described in Section 2.2, in combination with the multinomial group lasso predictor, see Section 2.4. The results obtained with and without using the domain adaption model where compared.

Primary tumors and metastases are biologically different, and it is therefore reasonable to expect a difference between their miRNA signature. Moreover technical differences between resections and core biopsies may influence the measured miRNA signature. It is therefore natural to expect that the performance of a predictor will differ between these two domains. We found, in our case, this difference to be fairly large.

We trained a multinomial group lasso predictor solely on the resections. This predictor achieved an overall expected generalization error of 11% on the resected primary tumors and benign liver samples, as estimated by 10-fold cross-validation. However, on liver core biopsies – which is our target domain – this predictor had an overall error of around 60%.

Note that all error rates reported henceforth in this paper originate from validation on our target domain. That is samples from one of the following categories: liver core biopsies of metastatic tumors, liver core biopsies of primary liver cancers or core biopsies of benign liver.

3.1 Prediction based on primary tumor samples

The results presented in this section were obtained using primary tumor and benign liver resections for training – i.e. no core biopsy samples where used for training. Error rates were estimated solely on liver core biopsy samples. We compared multinomial group lasso predictors obtained by training on one of the following three data sets:

a. Primary tumor and benign liver samples.
b. Artificial core biopsies obtained using the linear scale function.
c. Artificial core biopsies obtained using the log scale function.

For the simulation of artificial core biopsies, as described in Section 2.3, we used the 20 normal liver resections as contamination data. The distribution of α was taken as a beta distribution with shape parameters $(2, 2)$ for the 7 cancer classes and degenerate at 1 for the 2 benign liver classes. That is, for the benign liver classes no

contamination was added. The two artificial core biopsy data sets were generated by using either the linear or the log scale functions described in Section 2.2. The simulation was carried out with 750 samples from each class and the weights were chosen as

$$\omega_i = \frac{N_{\text{prim},y_i}}{750}$$

where N_{prim,y_i} is the number of primary samples of class y_i. For the two benign liver classes the simulation amounts to sampling with replacement from their empirical distribution. In practice, the simulation step for these two classes was therefore skipped and the 37 benign liver resections were just included, all with weight 1.

Figure 3 shows estimates of the expected generalization error against the number of miRNAs included in the predictor. A larger number of miRNAs corresponds to a lower value of the tuning parameter λ. The predictors trained directly on the primary cancer samples performed poorly when applied to the core biopsy samples. The best predictors achieved an overall error around 60%. Figure 4 further shows that effectively only the liver related classes got predicted correctly by these predictors. The overall error dropped to around 50% for the predictors trained on the artificial core biopsies using the linear scale function. Using the log scale function the overall error could be further reduced to around 35%. Figure 4 shows that the overall improvements embraced notable differences between sample subgroups. In particular, the log scale function performed best on core biopsies with a low tumor content, while it did not improve the error rate as much for the high tumor content samples. In fact, the linear scale function resulted in marginally better predictors for the high tumor content samples.

3.2 Prediction based on metastatic tumor samples

For comparison we trained a predictor solely on the available core biopsy samples, i.e. on metastatic cancer, primary liver cancer and benign liver samples. Using leave-one-out cross-validation the overall expected generalization error was estimated to about 55% for a predictor using around 100 miRNAs. It is likely that this error rate would be reduced if additional core biopsies where available for training.

3.3 Prediction based on primary and metastatic tumor samples

To further improve the predictors we investigated the effect of training on either primary tumor samples or artificial core biopsies in combination with available core biopsies. This was investigated in a setup where we included $n = 1, 2, 3$ or 4 core biopsy samples from each class in the training data.

To assess the performance we used the following subsampling procedure repeated 100 times.

1. Randomly split the core biopsy samples in a training and a test data set such that each contain roughly half of the samples per class.
2. Randomly sample n core biopsies from the training data set for each of the metastatic cancer classes.
3. Train the multinomial predictor on the combined training data set (artificial core biopsies/primary tumor samples + biological core biopsies selected for training).

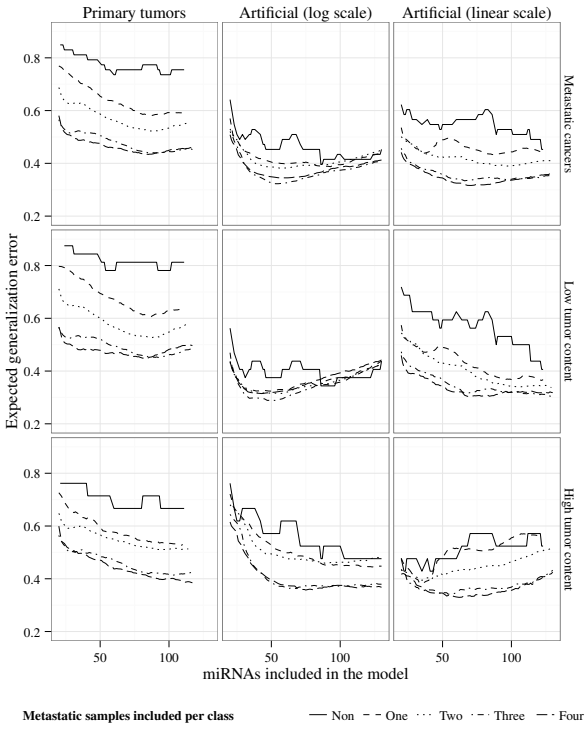


Fig. 5: Estimated expected generalization error for primary tumor prediction of metastatic liver core biopsies. The error is shown as a function of the estimated expected number of miRNAs included in the predictor. The plots show the error for predictors trained on primary tumor samples or artificial core biopsy samples in combination with biological core biopsies. The line type represents the number of biological core biopsies included per class in the estimation procedure. Biological core biopsies were not included in the hepatocellular carcinoma class, cholangiocarcinoma class and benign liver classes.

4. Estimate the misclassification error on the core biopsy test data set.

Figure 5 shows that the combination of primary tumors and core biopsies resulted in predictors with a smallest achievable error around 40% for the metastatic cancers. Combining the artificial core biopsies with the biological core biopsies this error was reduced to around 30% using the log or linear scale function. The optimal models were achieved with 40-60 miRNAs using the log scale and 60-80 miRNAs using the linear scale. Figure 5 further shows that for the log scale function no detectable improvement on the low tumor content samples was obtained by including biological core biopsies, while a considerable improvement was obtained for the high tumor content samples. For the linear scale function the improvement by including biological core biopsies is present for the low as well as the high tumor content samples.

4 DISCUSSION

Contamination of samples can affect the performance of molecular predictors. This has been established in other studies, e.g. Elloumi *et al*. (2011), and we have shown that the effect can be drastic in relation to primary tumor site identification. Other studies using molecular predictors for primary tumor site identification attempt to minimize benign tissue contamination by micro-dissection. However, micro-dissection may not always be applied to core biopsies or may cause delay in the diagnostic work-up. Hence tissue contamination remains to be an issue when core biopsy samples are involved.

We have developed a computational approach that deals with tissue contamination from surrounding tissue, and we have shown that the method drastically improves the performance of a molecular predictor on a non-trivial multiclass problem. The domain adaption approach, which we suggest, is flexible and yet simple to implement. In this paper we only considered metastatic liver core biopsies, but it is natural to assume that our contamination model works well for other biopsy sites than liver. In addition, although we only address tissue contamination specifically, our contamination model has the potential to be used to model background contamination in other types of samples.

Our results indicate that the suggested domain adaption approach was able to adjust for normal liver contamination present in core biopsy samples. By using this approach we were able to considerably reduce the error of primary tumor site identification for metastatic cancers. It is notable that inclusion of metastatic samples for training only resulted in a slight improvement of the error. Moreover this improvement may be caused by the additional number of independent biological samples included in the training set. This suggests that the contamination model successfully captures the liver contamination, and that there is not much more that can be learned about the target distribution.

Based on our findings, it is tempting to conclude that observed differences in molecular signatures from primary tumor resections and core biopsies of metastatic tumors are due to tissue contamination. However, our results do not support such a strong conclusion. Investigations of other differences at the molecular level between primary cancer and corresponding metastases may help to further improve molecular predictors.

Surprisingly, the improvements due to the contamination model were most pronounced for samples with a low tumor content. We would expect such samples to be more difficult to predict than high tumor content samples. The estimated tumor content is, in fact, an estimate of the relative amount of stroma and tumor content. Thus a possible explanation is potential stroma contamination, which is not part of our model.

ACKNOWLEDGEMENT

Funding: MV was supported by the Ministry of Science, Technology and Innovation, 09-049337. MV and NRH were supported by the University of Copenhagen Program of Excellence: *Statistical Methods for Complex and High Dimensional Models*.

REFERENCES

Albini, A., Mirisola, V., Pfeffer, U. (2008) Metastasis signatures: genes regulating tumormicroenvironment interactions predict metastatic behavior , *Cancer and Metastasis Reviews*, **27(1)**, 75-83.

Daumé,III, H., Marcu, D. (2006) Domain Adaptation for Statistical Classifiers , *Journal of Artificial Intelligence Research* , **26**, 101-126.

Elloumi, F., Hu, Z., Li, Y., Parker, J.S., Gulley, M.L., Amos, K.D., Troester, M.A. (2011) Systematic bias in genomic classification due to contaminating non-neoplastic tissue in breast tumor samples, *BMC Med Genomics*, **99**, 4:54.

Hastie, T., Tibshirani, R., Friedman, J. H. (2001) The elements of statistical learning: data mining, inference, and prediction

Lu, J., Getz, G., Miska, E.A., Alvarez-Saavedra, E., Lamb, J., Peck, D., Sweet-Cordero, A., Ebert, B.L., Mak, R.H., Ferrando, A.A., Downing, J.R., Jacks, T., Horvitz, H.R., Golub, T.R. (2005) MicroRNA expression profiles classify human cancers, *Nature*, **435(7043)** 834-8.

Mansour, Y., Mohri, M., Rostamizadeh, A. (2009) Domain Adaptation: Learning Bounds and Algorithms

Perell K., Vincent, M., Vainer, B., Petersen, B. L., Federspiel, B., Mller, A. K., Madsen, M., Hansen, N. R., Friis-Hansen, L., Nielsen, F. C., Daugaard, G. (2013) A microRNA-based primary tumor site classification of liver core biopsies. *submitted to Clinical Cancer Research*,

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J.P., Poggio, T., Gerald, W., Loda, M., Lander, E.S., Golub, T.R. (2001) Multiclass cancer diagnosis using tumor gene expression signatures, *Proc Natl Acad Sci U S A*, **98(26)** 15149-54.

Ramaswamy, S., Ross, K., Lander, E., Golub, T. (2002) A molecular signature of metastasis in primary solid tumors, *Nature Genetics*, **33**, 49-54.

Rosenfeld N, Aharonov R, Meiri E, Rosenwald S, Spector Y, Zepeniuk M, Benjamin H, Shabes N, Tabak S, Levy A, Lebanony D, Goren Y, Silberschein E, Targan N, Ben-Ari A, Gilad S, Sion-Vardy N, Tobar A, Feinmesser M, Kharenko O, Nativ O, Nass D, Perelman M, Yosepovich A, Shalmon B, Polak-Charcon S, Fridman E, Avniel A, Bentwich I, Bentwich Z, Cohen D, Chajut A, Barshack I. (2008) MicroRNAs accurately identify cancer tissue origin, *Nat Biotechnol*, **26(4)**, 462-9.

Vaerman, J L and Saussoy, P and Ingargiola, I (2004) Evaluation of real-time PCR data, *Journal of Biological Regulators and Homeostatic Agents* , **18**, 212-214.

VanGuilder, HD., Vrana, KE., Freeman, WM. (2008) Twenty-five years of quantitative PCR for gene expression analysis, *Biotechniques*, **44(5)**, 619-626.

Vincent, M., Hansen, N. (2012) Sparse group lasso and high dimensional multinomial classification (preprint) *ArXiv e-prints* 1205.1245

Chapter 7

Article: MicroRNAs predict primary tumor site

K. Perell, M. Vincent, B. Vainer, B. L. Petersen, B. Federspiel, A. K. Møller, M. Madsen, L. F. Hansen, N. R. Hansen, F. C. Nielsen, and G. Daugaard. A microrna-based primary tumor site classification of liver core-biopsies. *Clinical Cancer Research*

A microRNA-based primary tumor site classification of liver core biopsies.

Katharina Perell[1,2*], Martin Vincent[4*], Ben Vainer[3] ,Bodil Laub Petersen[3], Birgitte Federspiel[3], Anne Kirstine Møller[1], Mette Madsen[2], Niels Richard Hansen[4], Lennart Friis-Hansen[2], Finn Cilius Nielsen[2], Gedske Daugaard[1].

[1]Department of Oncology, [2]Center for Genomic Medicine and [3] Department of Pathology, Copenhagen University Hospital, Rigshospitalet, Blegdamsvej 9, DK-2100 Copenhagen Ø, Denmark. [4]Department of Mathematical Sciences, University of Copenhagen, DK-2100 Copenhagen Ø, Denmark.

*These first authors contributed equally to this work.

Running title*:* Primary tumor site classification of liver biopsies.

Key words: MicroRNA, classification, liver biopsy, metastases, surrounding tissue.

Financial support*:* This work has been supported in part by The Danish National Advanced Technology Foundation, The Danish Cancer Research Foundation, The Copenhagen University Hospital, Rigshospitalet, The Preben and Anna Simonsens Foundation and The Beckett Foundation.

Conflict of interest: No conflicts to disclose by any of the authors.

Corresponding Author: Katharina Perell,

Email: Katharina.Anne.Perell@regionh.dk,

Phone: +4520404564 or +4535459408

Fax: +4535455389

Translational relevance

Approximately 10-15 % of all cancer patients present with metastatic disease. Identification of the primary tumor site is clinically important, but remains a challenge. For the majority of these patients, primary tumor site identification relies on small biopsies from metastatic lesions. Tissue heterogeneity and inadequacy pose a problem for histopathological classification, but it is not yet established to which extend it affects molecular classification. We have developed a microRNA-based classifier, which predicts the primary tumor site of liver biopsies, using a minimum of tissue, with limited tumor content and without prior microdissection. Hence, molecular classification can proceed in concert with conventional diagnostic work-up, without compromising normal histopathological assessment. Our results indicate that surrounding normal tissue from the biopsy site may critically influence molecular classification. A significant improvement in classification accuracy was obtained when the influence of normal tissue was limited by application of a computational contamination model.

Abstract

Purpose*:* To develop a classifier based on microRNAs for primary tumor site identification of liver core biopsies and to explore the influence of surrounding normal liver tissue on classification.

Experimental Design*:* MicroRNA expression profiling was performed using quantitative Real-Time PCR on formalin-fixed paraffin-embedded samples. 278 primary tumors, liver metastases and normal liver samples were used as a training set, representing eight primary tumor classes, a class of squamous cell carcinoma (mixed population) and a class

of normal liver tissue. A computational model was applied to adjust for normal liver tissue contamination. Performance was estimated by cross-validation, followed by independent validation on 55 liver core biopsies representing metastases, primary liver cancer and normal liver tissue.

***Results**:* A microRNA classifier developed using the computational contamination model showed an overall classification accuracy of 74.5 % upon independent validation. Performance was estimated exclusively using small liver biopsies, with a tumor content as low as 10 %. For comparison, a classifier trained without adjusting for liver tissue contamination, showed a classification accuracy of 38.2 %.

***Conclusions**:* A clinically applicable microRNA classifier, which identifies the primary tumor site of liver core biopsies, was developed and validated. A potential risk of misclassification due to surrounding normal liver tissue was observed. By applying a computational contamination model, the classification accuracy improved significantly, obviating the need for microdissection.

Introduction

Current cancer treatment strategies are based on the anatomical site of the primary tumor. Therefore, a correct diagnosis of the primary tumor site remains an essential first step in disease management. Since more specific treatment regimens have emerged for many solid tumors, correct primary tumor site identification has become increasingly important. Despite improvements in imaging techniques and the use of immunohistochemical (IHC) markers, cancer patients presenting with metastatic disease at the time of diagnosis still represent a diagnostic challenge and in 3-5 % of these patients, the primary tumor site

remains undetectable [1]. As a result, these patients may be subjected to a time-consuming and expensive diagnostic work-up, resulting in treatment delay or even a suboptimal or incorrect treatment strategy.

In recent years, effort has been made towards establishing new supplementary diagnostic tools for primary tumor site identification. Molecular profiling is a promising diagnostic approach, which has the potential to provide an objective classification of uncertain or unknown metastatic cancers and render the diagnostic work-up of cancer patients more time- and cost-effective.

Several molecular classifiers, based on either messenger RNA (mRNA) or microRNA (miRNA) analysis, have been developed for primary tumor site identification. These classifiers show promising cross-validation and independent validation results. However, validation is often performed on a sample set predominantly constituted by primary tumors [2-7]. Primary tumors and their corresponding metastases may exhibit significant molecular differences due to altered biology or diversity in specimen sampling, which may influence classification accuracy. Such an influence may potentially be overlooked if metastatic samples represent a small part of the total validation set. Additionally, it is not well established to which extend normal tissue contamination affects molecular classification.

For the majority of patients with metastatic cancer, classification of the primary tumor site relies on formalin-fixed paraffin-embedded (FFPE) core biopsies from metastatic lesions. Standard specimen sampling methods result in heterogeneous samples, consisting of varying amounts of malignant cells and normal tissue [8]. A molecular classifier for primary tumor site identification in patients with metastatic disease must therefore be compatible with FFPE biopsy specimens, representing metastatic tissue with limited tumor content.

Furthermore, the possible influence on classification by normal tissue contamination must be considered. Essentially, the classifier performance must be assessed on representative samples for which the classifier is intended to perform.

The primary objective of this study was to develop a classifier able to identify the primary tumor site of FFPE liver core biopsies. Additionally, the classifier should be easy to apply in the daily clinic. Hence, the classifier should be able to perform on restricted tissue and tumor amount without the need for prior microdissection. We used miRNA, which is a class of small (21-24 nucleotides) non-coding RNA molecules [9], since these are known to be highly stable in FFPE tissue [10]. The biopsy site was limited to a single organ in order to explore the influence of surrounding normal tissue on primary tumor site classification. A computational contamination model was incorporated to allow classification of core biopsies even in the presence of normal liver tissue [11]. Furthermore we explored if the miRNA profile of metastases provides additional information necessary for correct classification, when compared to primary tumors.

Materials and Methods

Clinical samples

Tissue samples from 338 patients, corresponding to one of the following ten predefined assay classes, were obtained from archives of the pathology department, Copenhagen University Hospital, Rigshospitalet, Denmark: Lung cancer, breast cancer, gastric/cardia cancer, colorectal cancer, bladder cancer, pancreatic cancer, hepatocellular carcinoma, cholangiocarcinoma, squamous cell cancers of different origin, and normal liver tissue

The study was conducted according to national guidelines.

When selecting samples, the following issues were considered :(i) a single confident reference diagnosis was required. The reference diagnosis was established based on the original pathology report, clinical data and radiological findings; (ii) the training set should include the most common histological subtypes and represent a varied spectrum of dedifferentiation; (iii) each patient could only be represented by one sample, hence primary tumor samples and metastatic samples were unmatched.

Samples were formalin-fixed paraffin-embedded (FFPE) tissue specimens (dated 2000-2012). The sample set consisted of 199 surgical resections (162 primary tumors and 37 normal liver samples) and 134 liver core biopsies (109 primary liver cancers and liver metastases of known origin, and 25 normal liver samples). Normal liver samples were obtained from large surgical liver resections for colorectal metastases or from explanted livers. These samples were subdivided into (i) liver samples containing mild reactive changes due to the presence of a tumor in the proximity and (ii) cirrhotic liver. Cirrhosis was included in order to differentiate non-neoplastic fibrosis from the desmoplastic stromal reaction of metastatic lesions. Characteristics of the samples are shown in Table 1.

Primary tumors were assigned a differentiation grade, according to international guidelines. Additionally, all samples were independently reviewed by an expert pathologist to confirm the reference diagnosis and estimate the tumor percentage. The percentage of tumor tissue in resected samples (primary tumors) was defined as the relative amount of tumor cells. In core biopsies, the tumor tissue content was defined as the relative area of combined tumor tissue and desmoplastic stroma. The tumor percentage was estimated from a hematoxylin and eosin-stained section.

From each primary tumor resection, one 10 µm section was cut. To obtain tumor specific miRNA expression profiles, primary tumor samples were microdissected using the

Arcturus XT Microdissection System (Applied Biosystems, Foster City, CA) to ensure a tumor cell content of ≥ 60 %. The influence of non-malignant cells was limited by excluding samples with ≥ 50 % fibrosis, hemorrhage or necrosis (arbitrary cut-off).

Two sections of 5 µm were cut from each liver core biopsy, according to standard pathological procedures. No microdissection was performed on these samples. The only requirement was a minimum of 10 % tumor tissue without further limitations, regarding fibrosis, hemorrhage or necrosis.

Samples were initially split into a training set consisting of the 199 surgical resections and 79 core biopsies (2-12 biopsies in each class) and a validation set consisting of the remaining 55 liver core biopsies (5 samples randomly chosen from each class).

RNA extraction

Total RNA was extracted from FFPE tissue using a combination of ReCover All Total Nucleic Acid Isolation Kit (Ambion, Austin, Tx) and RNAqueous Micro Kit (Ambion). Briefly, the microdissected sections were deparaffinized by first adding 1 ml 100 % xylene and subsequently 1 ml 100 % ethanol. The later RNA extraction steps were similar for all dissected and non-dissected samples. The tissue was digested using 100 µl digestion buffer and 4 µl ProteinaseK (ReCover All) at 50 ˚C for 15 min and 80 ˚C for 15 min according to the manufacturer`s instructions. RNA was subsequently purified on columns and eluted in 15 µl elution solution (RNAqueous) according to the manufacturer`s protocol. Total RNA yield and quality was evaluated using Nanodrop ND-1000 spectrophotometer (NanoDrop Technologies, Wilmington).

miRNA Quantitative real-time PCR profiling

TaqMan low density array (TLDA) cards, human MicroRNA array A (Applied Biosystems) were used to determine the expression of 377 microRNAs. Each array contains six positive controls. RT-PCR reactions were performed according to the manufacturer's instructions. All reagents were obtained from Applied Biosystems. Briefly, 30 ng of total RNA was reverse transcribed (RT) with Megaplex RT primer human pool A and the TaqMan miRNA reverse transcription kit in a total volume of 7.5 µl per reaction. The amount of miRNA that can be extracted from core biopsies may be limited, and a 40 round pre-amplification step was therefore included. cDNA of liver core biopsies as well as primary tumor resections was pre-amplified in order to make the analysis from primary tumors and metastases comparable. Pre-amplification was performed using 2.5 µl RT product together with Megaplex PreAmp Primers and TaqMan PreAmp Master Mix in a 25 µl PCR reaction. Given that the FFPE tissue was not collected with RNA preservation in mind, we used the expression of the small nucleolar RNA, RNU44, as a surrogate measure of RNA integrity prior to miRNA quantification. RNU44 expression was examined in triplicate in the pre-amplified cDNA-mix using a 384-well plate. Each reaction consisted of 2.5 µl pre-amplified cDNA, 0.5 µl TaqMAn MicroRNA assay, 5 µl Universal Master Mix No AmpErase UNG and 2 µl Nuclease-free water (all from Applied Biosystems). PCR reactions were run on an Applied Biosystems 7900 HT system, according to the manufacturer`s instructions and analyzed using SDS software (v.2.4, automatic baseline setting). Based on preliminary results, we defined a mean cycle threshold (Ct) value ≤ 20 as a cut-off and only samples with Ct values below this cut-off were further processed. Pre-amplified cDNA was diluted with 375 µl 0.1 TE (ph. 8.0) and transferred to a TaqMan Human MicroRNA A array (v. 2.0). Quantitative real-time PCR was performed using an

Applied Biosystems ViiA 7 Real Time PCR and TaqMan Advanced Master Mix with 50 µl input cDNA template per lane. Ct values were calculated using the ViiA 7 software (v. 1.1). Successful analysis was performed for 333 samples (98.5 %). Five samples were excluded because the RNU44 Ct-values were above the predefined cut-off.

Statistical models and methods

Three different classifiers were developed based on the multinomial model for classification. The multinomial models were trained using a variation of multinomial lasso [12]. For two of the classifiers a contamination model was used to adjust for normal liver contamination. This adjustment was done by a simulation approach as part of the multinomial model training procedure [11]. The following procedures for data preprocessing, simulation and training were used:

1. Controls and miRNAs not expressed in the primary tumor samples were removed.
2. A simulated data set mimicking liver core biopsies was constructed from the contamination model, using only normal liver and primary tumor resections from the training set.
3. All samples were normalized by centering and scaling the individual samples to mean 0 and variance 1.
4. For each of the three classifiers the multinomial model was trained using the training data, the simulated data or a combination of the two data sets.

To model miRNA expression of liver contaminated core biopsies (step 2), a simulated data set was constructed using a contamination model [11]. The contamination model is specified as a mixture of miRNA expressions from primary tumor and normal liver according to

$$\alpha \times \text{primary tumor signature} + (1 - \alpha) \times \text{normal liver signature}$$

where α denotes the tumor percentage. The tumor percentage was taken to follow a beta distribution with first shape parameter equal to four and second shape parameter equal to three. Due to non-linearity of the PCR amplification, the model was not applied to the observed scale but to a suitably transformed scale [11]. Random sampling with replacement of the primary tumor and the normal liver samples in the training data set as well as random sampling of α was used to simulate core biopsy samples by the contamination model.

For classifier development, we used the multinomial group lasso model as previously described [13]. In our set-up, the multinomial model is a model of the probability of the 10 assay classes given the observed 377 miRNA expression measurements from each sample. The log-probability of each class is, up to a constant, a weighted sum of the miRNA expressions. The model provides an estimate of the class probability and not just a classification. As a consequence of the multinomial group lasso method, the weights for some miRNAs will be 0 for all classes. The method thus automatically selects those miRNAs, most relevant for classification. Standardization of miRNA expressions across samples was done internally in the training algorithm to avoid that difference in scale could influence the miRNA selection.

The multinomial group lasso method produce a sequence of 100 models, with each model selecting different combinations of miRNAs. To select a final model, from the 100 produced models, an additional model selection procedure was performed. This was done by cross-validation using the negative log-likelihood loss.

To obtain an unbiased assessment of the performance of the final model we used nested cross-validation. By an outer cross-validation loop we estimated the performance of the

model obtained by the combined training and cross-validation based model selection procedure. In addition to the cross-validation an independent validation was performed.

Results

MicroRNA classifier based exclusively on primary tumors misclassifies core biopsies.

To investigate whether a miRNA profile obtained exclusively from primary tumors was able to classify the primary tumor site of liver core biopsies, predominantly consisting of metastases, we subdivided our training set. The original training samples were divided into a smaller training set consisting of the 199 resections (primary tumors and normal liver samples) and a test set consisting of the 79 liver core biopsies (metastases, primary liver cancer and normal liver tissue). This resulted in a model based on expression profiles from 55 miRNAs (PRIM classifier). The PRIM classifier showed a 90 % overall accuracy upon 10-fold cross validation (Supplementary Figure S1). When applied to the 79-core biopsy test set, the accuracy dropped to 44.3 % (Table 2) with a pronounced difference in classification accuracy across the different assay classes. The PRIM classifier performed well on core biopsies consisting of normal liver, but generally poor on metastases from non-liver derived primary tumors. Liver metastases from colorectal cancer constituted an exception, with 67 % being classified correctly. A complete list of classifier predictions is given in Supplementary Table S1. Approximately 40 % (17/43) of the misclassified samples were classified as normal liver (reactive liver or cirrhosis) and 35/40 misclassified metastases from non-liver derived primary tumors were classified as either primary liver cancer or normal liver. This strongly indicated that contamination with normal liver in core

biopsies impeded correct classification. A principal component plot (Figure 1) illustrates how core biopsies independent of class clustered together with liver derived samples.

Application of a contamination model for classifier training improves classification of core biopsies.

To improve classification of liver core biopsies, we used a computational contamination model (CCM). Based on the assumption that the level of individual miRNAs in tumor tissue and the surrounding liver tissue is independent of one another, samples constructed using the contamination model mimic liver core biopsies. These samples were constructed for each assay class only using miRNA profiles of normal liver and primary tumor resections from the 199-sample training set. By exchanging the original primary tumors with the computational constructed samples as a training set, we developed a model consisting of 104 miRNAs (CCM classifier).

To test the performance of the CCM classifier on liver core biopsies, we applied the 79 liver core biopsy test set. The accuracy showed an improvement across most assay classes, with a pronounced effect on non-liver derived malignancies, resulting in an overall accuracy of 58.2 % (Table 2). The improved classification accuracy was largely due to a reduction in samples being misclassified as normal liver (8/32) and fewer metastases from non-liver derived primary tumors being misclassified as derived from the liver (11/30) (see Supplementary Table S1 for a complete list of classifier predictions).

Liver biopsies may feature important information for correct classification.

miRNA signatures may differ between primary tumors and metastases, not only due to normal tissue contamination but also due to underlying biological differences. Such biological differences will obviously not be present in the computational constructed

samples. Therefore, to encompass a potential molecular difference between primary tumors and metastases, we used the same computational developed training samples as described for the CCM classifier together with the 79 liver core biopsies (CB). From this combined training set, we developed a model consisting of 116 miRNAs (CCM+CB classifier). To estimate the performance of this CCM+CB classifier, 8-fold cross-validation was performed, which showed 67.1 % overall accuracy (Table 2 and Supplementary Figure S2). Further, an independent validation using 55 liver core biopsies was performed, demonstrating an overall accuracy of 74.5 %. Figure 2 shows the independent validation results of the CCM+CB classifier illustrated by a confusion matrix. For comparison, we applied the independent validation set to the PRIM classifier and the CCM classifier and obtained overall accuracies of 38.2 % and 67.3 %, respectively. A comparison of validation results between the three classifiers is shown in Table 3.

An important feature of a clinical applicable classifier is the ability to deliver a single high confident prediction. By proposing two or more differential diagnosis, uncertainty and subjectivity may be imposed. Due to these considerations, training biopsies were used to establish a threshold for high-confidence predictions. Based on this threshold a prediction was defined as high confidence if the model probability was larger than 0.6. When applied to the independent validation set, 65 % of the samples were high-confidence predictions, with 89 % being classified according to the reference diagnosis. Table 4 shows the number of high confidence predictions in the independent validation set.

The miRNAs included in the 55-microRNA PRIM classifier, the 104-microRNA CCM classifier and the 116-microRNA CCM+CB classifier are listed in Supplementary Table S2**.** Forty-two miRNAs were included in all three classifiers, whereas 5 miRNAs were only represented in the PRIM classifier, amongst them miR-122, known to be specifically

expressed in liver tissue. The additional miRNAs in the CCM and CCM+CB classifiers tend to reflect contribution of more miRNAs belonging to the same families.

Subclass analysis.

Approximately 40 % of the core biopsies included in this study yielded a tumor percentage lower than 50%, with 26 % of the samples containing ≤ 35 % tumor tissue. To address whether a low tumor percentage could compromise classification, an analysis based on the amount of tumor tissue in the samples was performed. Liver core biopsies containing tumor tissue were subdivided into low (≤ 35 %), moderate (>35-65 %) and high tumor content (> 65 %). We observed a slightly higher overall error rate for samples with low tumor content (13/31 samples), while no difference in error rate was observed for samples yielding moderate (10/33 samples) or high (14/45 samples) tumor content. Supplementary Table S1 provides a complete list of predictions for each of these classifiers together with tumor percentage and sample age.

Discussion

We present the development and validation of a microRNA (miRNA) classifier, designed as a supplementary diagnostic tool to histopathological evaluation and imaging, during the diagnostic work-up of patients suspected of malignant liver disease (i.e. metastases or primary liver cancer). The classifier is trained on primary tumors, liver metastases and normal liver tissue and consists of expression profiles from 116 miRNAs. The classifier performs on formalin-fixed paraffin-embedded (FFPE) liver core biopsies with limited amount of tumor tissue and varying amount of normal liver tissue. It distinguishes between eight primary tumor classes, squamous cell carcinoma (mixed population) and normal liver tissue with an overall accuracy of 67.1 % upon cross-validation and 74.5 % upon an

independent validation. To mimic the daily diagnostic routine, we validated the classifier using small sections of liver core biopsies with as little as 10 % tumor tissue and refrained from microdissection. This allows miRNA analysis to proceed independently and simultaneously with the histopathological work-up, without causing delay in the final diagnostic subgrouping of the patient.

As opposed to previously reported classifiers, we limited the application to a single biopsy site. This allowed us to study the impact of surrounding normal tissue on primary tumor site classification, avoiding classification bias caused by biopsy site and reducing the number of validation samples needed. The liver was chosen because: (i) it is a common site for metastatic disease and the most common single site of metastatic involvement in patients with carcinoma of unknown primary site (CUP) [1]; (ii) it represents the most common metastatic site for gastrointestinal (GI) cancers [14] and (iii) the liver is easily accessible for biopsy.

Tumor samples contain varying amounts of malignant cells, stromal cells and surrounding (contaminating) normal tissue from the biopsy/resection site. The influence of surrounding normal tissue on molecular classification is not clear, although a potential systematic classification bias, caused by normal tissue, has been reported [15, 16]. Most previously developed diagnostic classifiers require high tumor content (≥60 % tumor) and use microdissection on selected samples, prior to gene expression analysis, for tumor cell enrichment. Although microdissection reduces the surrounding normal tissue, it also holds several disadvantages. Most importantly, microdissection may not always be possible due to a relatively small number of tumor cells located dispersedly in the liver core biopsy [8]. In addition, microdissection may be time consuming. To investigate the influence of normal liver tissue contamination, we constructed a miRNA classifier (PRIM classifier) only based

on primary tumor samples. Although a high cross-validation accuracy of 90 % was achieved, the PRIM classifier showed a disappointing classification accuracy of 38.2 % on the independent validation set consisting of liver core biopsies. The low classification accuracy was predominantly caused by samples being misclassified as normal liver or primary liver malignancies. By adjusting for normal liver contamination, the accuracy improved significantly to 67.3 % upon independent validation. Hence, our results indicate that a miRNA signature is sustained in metastases compared to corresponding primary tumors, but contamination with surrounding normal tissue must be considered a potential cause of error in molecular primary tumor site classification based on miRNA.

The genetic events responsible for the metastatic process are still poorly understood. The key question is whether metastasis is driven by mutations that occur after the tumor cells arrive at a distant site [17] or whether the complete repertoire of somatic mutations are generated by heterogeneous clones in the primary tumor [18, 19]. Increasing evidence support a key role for miRNAs in cancer cell invasion, migration and metastasis [20], and studies have reported altered miRNA signatures in metastases compared to matched primary tumors [21, 22]. By including metastatic liver core biopsies in the training set, we observed an increase in classification accuracy from 67.3 % to 74.5 % upon independent validation, with a notable effect on pancreatic cancer classification. This improvement could be due to different genetic information in metastases, compared to primary tumors. Metastases from the GI tract, especially pancreatic and gastric cancers, are usually difficult to distinguish from one another and from cholangiocarcinomas by histopathology alone [23, 24]. As illustrated in Supplementary Fig. S2, our classification demonstrates a similar tendency, especially upon cross-validation. The tendency was less clear from the independent validation results (Fig 2). Still, a significant proportion of GI tract cancers

were correctly classified from their miRNA signature. Cholangiocarcinomas constituted an exception, since only 1 out of 5 validation samples was correctly classified. The poor classification accuracy was predominantly due to misclassification as normal liver. Samples with moderate to high tumor content showed no difference in classification rate, but a trend towards a slightly higher error rate in samples with low tumor content was observed. We defined and estimated tumor content in core biopsies as both tumor- and stromal-cells, which may explain why no further improvement in classification was observed for samples with high tumor content. Due to differences in class distribution and an unequal number of samples, the impact of tumor percentage on classification must be interpreted with caution. Still, approximately 60 % of all samples with low tumor content were correctly classified.

In recent years, two diagnostic mRNA classifiers [4, 25] and one miRNA classifier [3] have become commercially available. It is difficult to compare performance across classifiers, due to different configurations of assay classes. Obviously, differences in number and sample distribution among the included assay classes affect the performance estimate, but other important considerations need to be highlighted. First, the performance of most classifiers is often estimated on a combination of primary tumor samples and metastatic samples, with metastases contributing merely 1/3 of the total validation set. In the present study, we showed that primary tumor site classification from primary tumor samples reached 90 % accuracy but only 38.2 % accuracy was achieved when the classifier was applied to an independent set of liver core biopsies, predominantly constituting metastases. Second, most classifiers are validated on a combination of resections and biopsies. Since the relative amount of normal surrounding tissue usually is higher in biopsies than in resections, the influence of normal tissue contamination may be

overlooked by this approach. The importance of validating a molecular classifier on representative samples on which it is intended to perform, remains essential in order to avoid potential overestimation of classifier performance.

Ideally, a classifier should be able to distinguish between every known primary tumor, and every known subtype. However, this may not be possible due to overlap between genetic signatures and limited number of samples. Therefore, the selection of primary tumor classes in the present study was made to encompass (i) primary tumors that often metastasize to the liver, (ii) primary tumors difficult to diagnose with conventional diagnostic methods and (iii) common primary tumors for which an effective systemic treatment is available, making a correct tumor classification clinically important. Liver metastases may originate from primary tumors not included in the training set. This is not different from other classifiers, but highlights the importance of a multidisciplinary approach in cancer diagnostics.

Histopathology remains the cornerstone in primary tumor site identification. Although diagnostic accuracy has improved with the routine use of IHC markers, the primary tumor site is still missed in a substantial number of patients with metastatic cancer, due to unspecific morphological appearance and lack of specific IHC markers. Furthermore, diagnosis of the primary tumor site of metastatic tissue by histopathology often requires a step-wise and time-consuming approach [26]. An important and yet unanswered question is whether molecular classification adds to the preexisting diagnostic work-up. In a recently published meta-analysis, the classification accuracy of immunohistochemistry was estimated based on five studies, showing an expected mean accuracy of 65.6 % in primary tumor site identification of metastatic cancers [27]. However, the results may not reflect the true ability of IHC, due to differences in class distribution among the included

studies and a restricted number of IHC stains. Importantly, 3 out of 5 studies in the meta-analysis either did not include GI cancers or included GI cancers as one common group [28-30]. A comparison between molecular classification and IHC guided methods was recently reported in a pilot study, showing a higher percentage of correctly diagnosed metastatic samples, when a molecular classification method was applied [31]. Notably, a substantial inter-observer difference was revealed amongst the five pathologists who took part in the study. Hence, the multi-disciplinary diagnostic work-up of cancer patients, which includes clinical assessment, imaging and histopathology, may indeed improve if molecular classification is added.

In conclusion, we have developed a miRNA classifier, which is able to determine the primary tumor site of FFPE liver core biopsies. Based on our data set, the signal provided by the surrounding normal liver hampered correct classification significantly. By applying a computational contamination model, adjustment of the liver signal was accomplished and a valid classification could be established on tissue containing less than 35 % tumor, making prior microdissection redundant. The results of an independent validation study performed entirely on liver core biopsies, predominantly representing metastatic tumors, is encouraging. Due to a limited amount of core biopsies, the independent validation was performed on a restricted data set, which is a limitation of our study. Notably, the validation samples reflect the characteristics of the underlying population of interest. The classifier was designed to perform on biopsies exclusively from the liver. Although it has several advantages, it also narrows the clinical application. In order to broaden the future use of the classifier, it seems reasonable to believe that the classifier could be transformed, enabling diagnosis of metastases from several other sites including lymph nodes, lung and bone.

In patients where IHC provides inadequate classification (none or multiple diagnoses) or when histopathology and imaging offers discordant diagnostic suggestions, the miRNA classifier may add important diagnostic information. An independent validation on a larger sample set consisting of metastatic lesions and a prospectively conducted study is planned to validate the applicability of this classifier in the diagnostic setting.

Authors Contributions

Conception and design: K. Perell, M. Vincent, B. Vainer, B. L. Petersen, B. Federspiel, A. K. Møller, F. C. Nielsen, G. Daugaard.

Development and methodology: K. Perell, M. Vincent, L. Friis-Hansen, M. Madsen, A. K. Møller, F. C. Nielsen, G. Daugaard.

Acquisition of data: K. Perell, B. Vainer, B. L. Petersen, B. Federspiel.

Analysis and interpretation of data: K. Perell, M. Vincent.

Writing, review and/or revision of the manuscript: K. Perell, M. Vincent, B. Vainer, B. L. Petersen, B. Federspiel, M. Madsen, A. K. Møller, L. Friis-Hansen, N. R. Hansen, F. C. Nielsen, G. Daugaard.

Administrative, technical or material support: K. Perell, M. Madsen, L. Friis-Hansen.

Study supervision: N. Richard Hansen**,** G. Daugaard, F. C. Nielsen.

Review of pathology findings that assisted in selection of cases: B. Vainer, B. L. Petersen, B. Federspiel.

Acknowledgements

The authors wish to thank Ewa Futoma-Kazmierczak, Mette Hedegaard Moldaschl and Jonas Vikeså for technical support and Snjòlaug Nielsdottir, MD for pathological assistance.

Reference List

[1] Pavlidis N, Pentheroudakis G. Cancer of unknown primary site. Lancet 2012;379:1428-35.

[2] Ma XJ, Patel R, Wang X, et al. Molecular classification of human cancers using a 92-gene real-time quantitative polymerase chain reaction assay. Arch Pathol Lab Med 2006;130:465-73.

[3] Meiri E, Mueller WC, Rosenwald S, et al. A second-generation microRNA-based assay for diagnosing tumor tissue origin. Oncologist 2012;17:801-12.

[4] Pillai R, Deeter R, Rigl CT, et al. Validation and reproducibility of a microarray-based gene expression test for tumor identification in formalin-fixed, paraffin-embedded specimens. J Mol Diagn 2011;13:48-56.

[5] Ramaswamy S, Tamayo P, Rifkin R, et al. Multiclass cancer diagnosis using tumor gene expression signatures. Proc Natl Acad Sci U S A 2001;98:15149-54.

[6] Su AI, Welsh JB, Sapinoso LM, et al. Molecular classification of human carcinomas by use of gene expression signatures. Cancer Res 2001;61:7388-93.

[7] Talantov D, Baden J, Jatkoe T, et al. A quantitative reverse transcriptase-polymerase chain reaction assay to identify metastatic carcinoma tissue of origin. J Mol Diagn 2006;8:320-9.

[8] Cheng L, Zhang S, MacLennan GT, et al. Laser-assisted microdissection in translational research: theory, technical considerations, and future applications. Appl Immunohistochem Mol Morphol 2013;21:31-47.

[9] Finnegan EF, Pasquinelli AE. MicroRNA biogenesis: regulating the regulators. Crit Rev Biochem Mol Biol 2013;48:51-68.

[10] Hall JS, Taylor J, Valentine HR, et al. Enhanced stability of microRNA expression facilitates classification of FFPE tumour samples exhibiting near total mRNA degradation. Br J Cancer 2012;107:684-94.

[11] Vincent M, Perell K, Nielsen FC, Daugaard G, Hansen NR. Modeling tissue contamination to improve molecular identification. Submitted to Bioinformatics May 2013.

[12] Friedman J, Hastie T, Tibshirani R. Regularization Paths for Generalized Linear Models via Coordinate Descent. J Stat Softw 2010;33:1-22.

[13] Vincent M, Hansen NR. Sparse group lasso and high dimensional multinomial classification. Computational Statistics and Dataanalysis, under revision.

[14] Hess KR, Varadhachary GR, Taylor SH, et al. Metastatic patterns in adenocarcinoma. Cancer 2006;106:1624-33.

[15] Elloumi F, Hu Z, Li Y, et al. Systematic bias in genomic classification due to contaminating non-neoplastic tissue in breast tumor samples. BMC Med Genomics 2011;4:54.

[16] Staub E, Buhr HJ, Grone J. Predicting the site of origin of tumors by a gene expression signature derived from normal tissues. Oncogene 2010.

[17] Shah SP, Morin RD, Khattra J, et al. Mutational evolution in a lobular breast tumour profiled at single nucleotide resolution. Nature 2009;461:809-13.

[18] Ding L, Ellis MJ, Li S, et al. Genome remodelling in a basal-like breast cancer metastasis and xenograft. Nature 2010;464:999-1005.

[19] Yachida S, Jones S, Bozic I, et al. Distant metastasis occurs late during the genetic evolution of pancreatic cancer. Nature 2010;467:1114-7.

[20] Baranwal S, Alahari SK. miRNA control of tumor cell invasion and metastasis. Int J Cancer 2010;126:1283-90.

[21] Chen W, Tang Z, Sun Y, et al. miRNA expression profile in primary gastric cancers and paired lymph node metastases indicates that miR-10a plays a role in metastasis from primary gastric cancer to lymph nodes. Exp Ther Med 2012;3:351-6.

[22] Gravgaard KH, Lyng MB, Laenkholm AV, et al. The miRNA-200 family and miRNA-9 exhibit differential expression in primary versus corresponding metastatic tissue in breast cancer. Breast Cancer Res Treat 2012;134:207-17.

[23] Oien KA. Pathologic evaluation of unknown primary cancer. Semin Oncol 2009;36:8-37.

[24] Park SY, Kim BH, Kim JH, Lee S, Kang GH. Panels of immunohistochemical markers help determine primary sites of metastatic adenocarcinoma. Arch Pathol Lab Med 2007;131:1561-7.

[25] Erlander MG, Ma XJ, Kesty NC, Bao L, Salunga R, Schnabel CA. Performance and clinical evaluation of the 92-gene real-time PCR assay for tumor classification. J Mol Diagn 2011;13:493-503.

[26] Oien KA, Dennis JL. Diagnostic work-up of carcinoma of unknown primary: from immunohistochemistry to molecular profiling. Ann Oncol 2012;23 Suppl 10:x271-x277.

[27] Anderson GG, Weiss LM. Determining tissue of origin for metastatic cancers: meta-analysis and literature review of immunohistochemistry performance. Appl Immunohistochem Mol Morphol 2010;18:3-8.

[28] Brown RW, Campagna LB, Dunn JK, Cagle PT. Immunohistochemical identification of tumor markers in metastatic adenocarcinoma. A diagnostic adjunct in the determination of primary site. Am J Clin Pathol 1997;107:12-9.

[29] Gamble AR, Bell JA, Ronan JE, Pearson D, Ellis IO. Use of tumour marker immunoreactivity to identify primary site of metastatic cancer. BMJ 1993;306:295-8.

[30] Lagendijk JH, Mullink H, van Diest PJ, Meijer GA, Meijer CJ. Immunohistochemical differentiation between primary adenocarcinomas of the ovary and ovarian metastases of colonic and breast origin. Comparison between a statistical and an intuitive approach. J Clin Pathol 1999;52:283-90.

[31] Kulkarni A, Pillai R, Ezekiel AM, Henner WD, Handorf CR. Comparison of histopathology to gene expression profiling for the diagnosis of metastatic cancer. Diagn Pathol 2012;7:110.

Figure Legends

Figure 1. Principal Component Plot illustrating the clustering of training samples. Samples were divided into: **A**) Core biopsies from non-liver derived malignancies representing metastases from bladder, breast, colorectal, gastric/cardia, lung, pancreatic cancer and mixed primary tumors of squamous cell morphology. **B)** Resected non-liver derived malignancies constituting primary tumors from the same 7 classes mentioned above. **C)** Core biopsies from liver derived malignancies (hepatocellular carcinoma, cholangiocarcinoma) and normal liver (reactive and cirrhotic). **D)** Resected liver-derived malignancies and normal liver.

Figure 2. Confusion matrix showing CCM+CB classifier predictions upon the independent validation set consisting of 55 liver core biopsies. Each row and column corresponds to one of the assay classes included in the classifier. Columns indicate classes according to the reference diagnosis; rows indicate the diagnosis predicted by the CCM+CB classifier. Numbers on the diagonal indicate cases for which the predicted diagnosis matched the reference diagnosis, whereas off-diagonal numbers were in disagreement and counted as test errors. The positive percentage agreement for each class was calculated.
Squamous, Squamous cell carcinoma (mixed population); **CCA**, cholangiocarcinoma; **CRC**, colorectal carcinoma; **GC**, gastric or cardia carcinoma; **HCC**, hepatocellular carcinoma.

Supplementary Figure S1. Confusion matrix showing PRIM classifier predictions upon cross-validation. The PRIM classifier was trained on primary tumor and normal liver resections only. Classes according to reference diagnosis are shown along the columns and classes according to classifier predictions are shown along the rows. The positive percentage agreement for each assay class was calculated.
Squamous, squamous cell carcinoma (mixed population); **CCA**, cholangiocarcinoma; **CRC**, Colorectal carcinoma; **GC**, gastric or cardia carcinoma; **HCC**, hepatocellular carcinoma,

Supplementary Figure S2. Confusion matrix showing CCM+CB classifier predictions upon 8-fold cross-validation. Cross-validation was performed on 79 liver core biopsies. Each assay class was represented by 2-12 samples. Each row and column corresponds to one of the assay classes included in the classifier. Columns indicate classes according to

the reference diagnosis; rows indicate the diagnosis predicted by the CCM+CB classifier. Numbers on the diagonal indicate cases for which the predicted diagnosis matched the reference diagnosis, whereas off-diagonal numbers were in disagreement and counted as test errors. The positive percentage agreement for each class was calculated. **Squamous**, squamous cell carcinoma (mixed population); **CCA**, cholangiocarcinoma; **CRC**, colorectal carcinoma; **GC**, gastric or cardia carcinoma; **HCC**, hepatocellular carcinoma.

Table 1.Selected characteristics of samples included in classifier training and validation

Tissue of origin	Histology	Resection no. (TR)	Biopsy no. (TR)	Biopsy no. (V)
Bladder	Urothelial carcinoma	17	2	5
Breast	Invasive ductal, lobular, medullar	17	7	5
Billiary tract	Adenocarcinoma	20	4	5
Colorectal	Adenocarcinoma, mucinous adenocarcinoma	20	12	5
Gastric/cardia	Adenocarcinoma, signet ring cell carcinoma	18	12	5
Liver	Hepatocellular carcinoma	17	3	5
Normal liver	Reactive	20	7	5
Normal liver	Cirrhotic	17	8	5
Lung	Adenocarcinoma, Mixed type, Large cell	17	2	5
Pancreas	Adenocarcinoma	20	10	5
Cervix, Lung, Anal, Esophagus, Head and Neck	Squamous cell carcinoma	16	12	5
Total		199	79	55

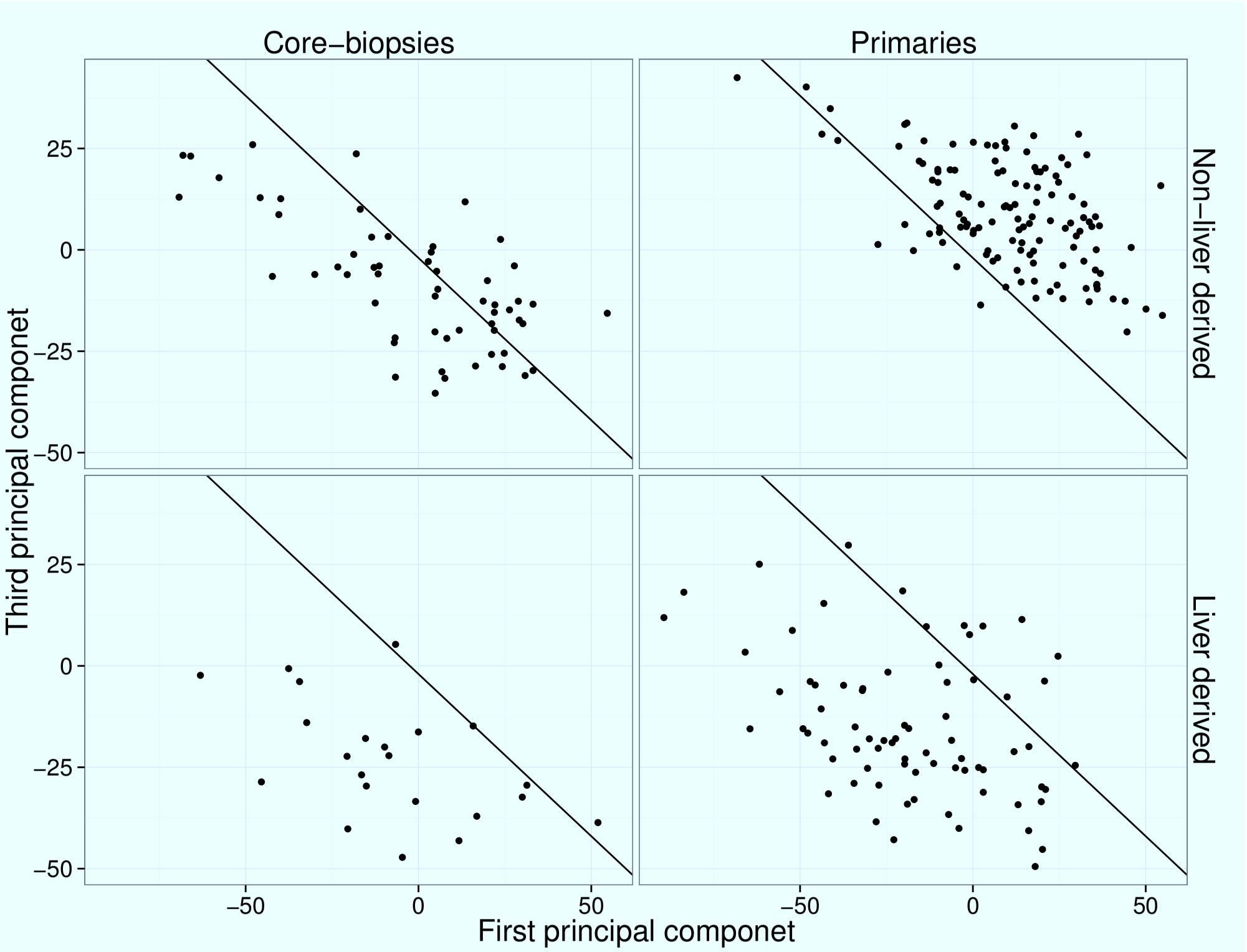
The tissue of origin, histology and number (no.) of samples used for classifier training (TR) and independent validation (V) are listed. Normal liver was subdivided into reactive and cirrhotic liver, but was regarded as one class. Squamous cell carcinoma was regarded as one class of mixed population.

Resection, primary tumor and normal liver resections; **Biopsy**, liver core biopsies consisting of liver metastases, primary liver cancer and normal liver.

Table 2. Performance of the PRIM classifier, CCM classifier and CCM+CB classifier on the 79-core biopsy sample set.

	Reference site										
Classifier	Bladder (2)	Breast (7)	CCA (4)	CRC (12)	GC (12)	HCC (3)	Lung (2)	Pancreas (10)	Squamous cell (12)	Normal liver (15)	**Overall accuracy**
PRIM	0	2	4	8	2	1	1	1	2	14	44.3 %
CCM	0	0	2	9	8	1	1	3	8	14	58.2 %
CCM+CB	1	4	4	8	8	1	1	4	9	13	67.1 %

Each assay class was represented by 2-15 samples, as marked in brackets. The number of correctly classified samples according to the reference diagnosis is listed for each assay class. The sample set constituted a test set for the PRIM and CCM classifier. For the CCM+CB classifier, performance was estimated by eight-fold cross validation.
Squamous, squamous cell carcinoma (mixed population). **Normal liver**, 8 cirrhotic and 7 reactive liver samples. **CCA**, cholangiocarcinoma; **CRC**, colorectal carcinoma; **GC**, gastric or cardia carcinoma; **HCC,** hepatocellular carcinoma.


Core−biopsies
Primaries
Non−liver derived
Liver derived
Third principal componet
First principal componet
25
0
−25
−50
−50
0
50

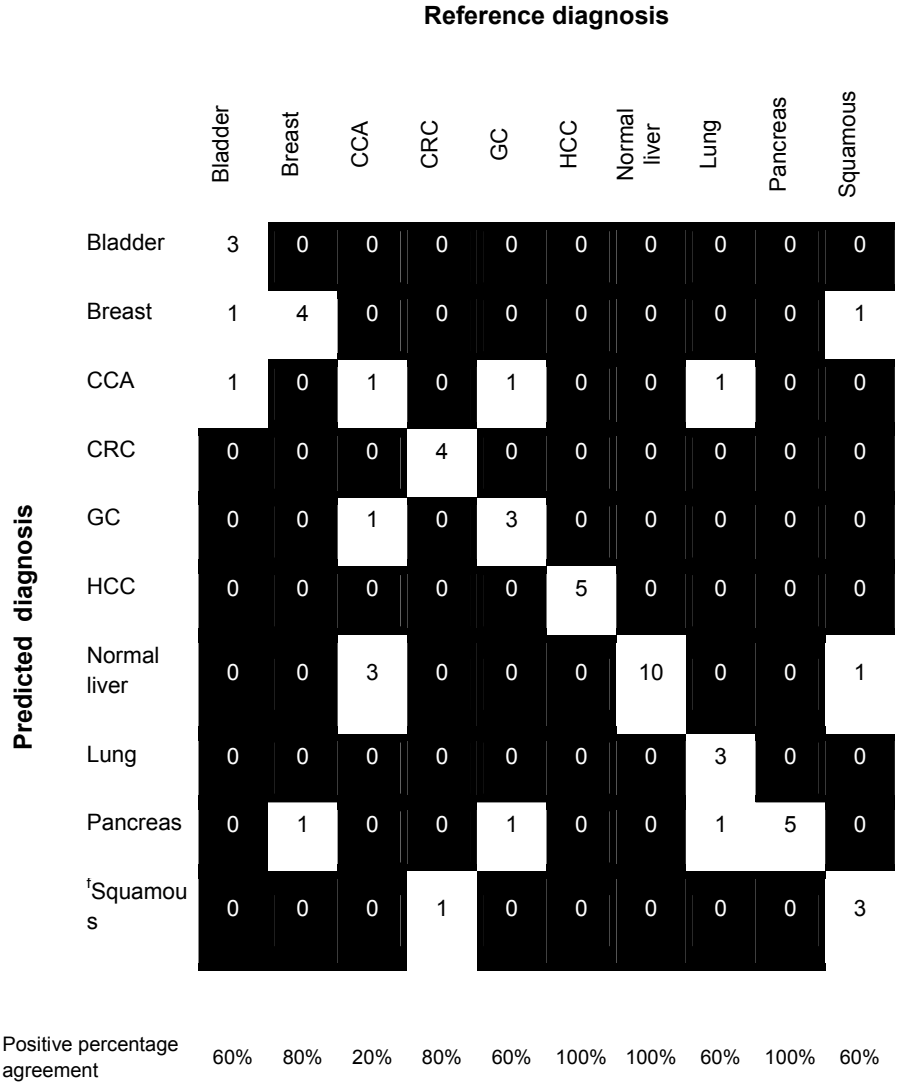

Reference diagnosis
Predicted diagnosis
Bladder
Breast
CCA
CRC
GC
HCC
Normal liver
Lung
Pancreas
Squamous
Bladder 3 0 0 0 0 0 0 0 0 0
Breast 1 4 0 0 0 0 0 0 0 1
CCA 1 0 1 0 1 0 0 1 0 0
CRC 0 0 0 4 0 0 0 0 0 0
GC 0 0 1 0 3 0 0 0 0 0
HCC 0 0 0 0 0 5 0 0 0 0
Normal liver 0 0 3 0 0 0 10 0 0 1
Lung 0 0 0 0 0 0 0 3 0 0
Pancreas 0 1 0 0 1 0 0 1 5 0
†Squamous 0 0 0 1 0 0 0 0 0 3
Positive percentage agreement 60% 80% 20% 80% 60% 100% 100% 60% 100% 60%

Figure 2.

Table 3. Results of the independent validation of the PRIM classifier, CCM classifier and CCM+CB classifier.

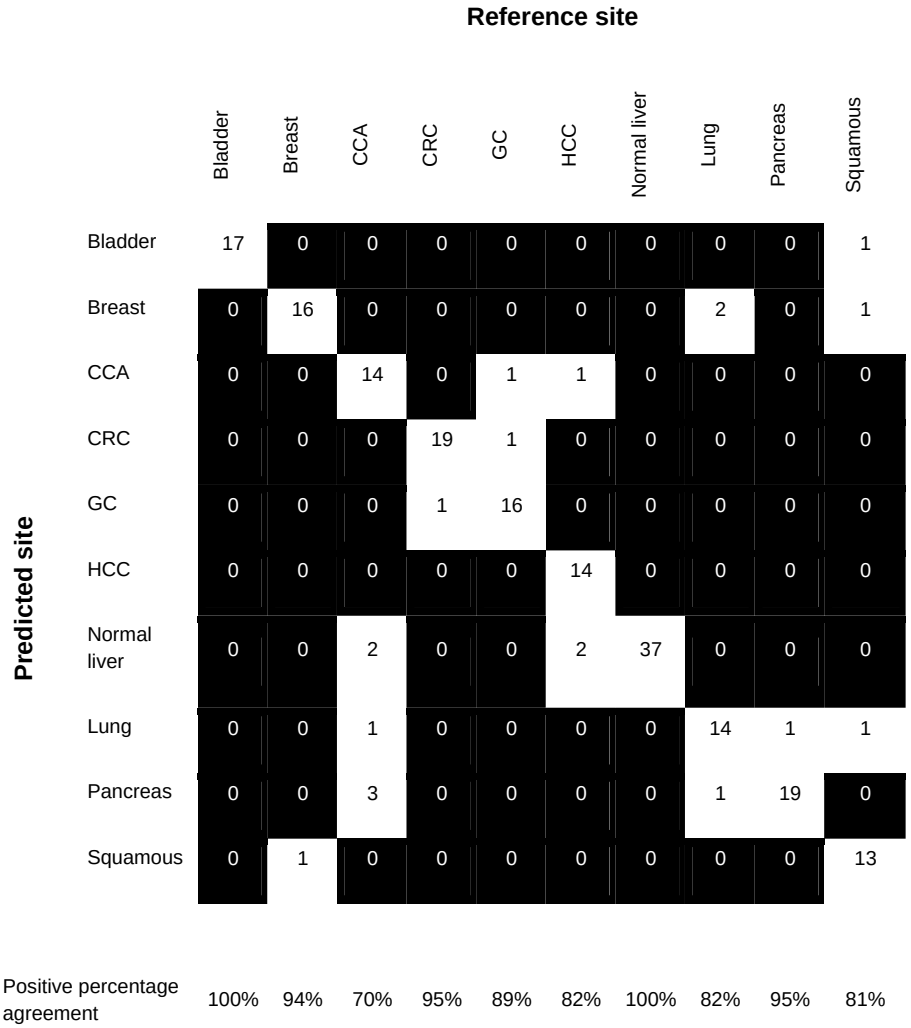
	Reference site										
Classifier	Bladder (5)	Breast (5)	CCA (5)	CRC (5)	GC (5)	HCC (5)	Lung (5)	Pancreas (5)	Squamous cell (5)	Normal liver (10)	**Overall accuracy**
PRIM	1	1	1	3	1	3	0	0	2	9	38.2 %
CCM	3	4	3	4	2	4	3	0	4	10	67.3 %
CCM+CB	3	4	1	4	3	5	3	5	3	10	74.5 %

The performance of each of the three classifiers on the independent validation set consisting of 55 liver core biopsies is shown. Each class was represented by 5 samples, except the Normal liver class, which consisted of 5 reactive liver samples and 5 cirrhotic liver samples. The number of correctly classified samples according to the reference diagnosis is listed for each assay class. **Squamous**, squamous cell carcinoma (mixed population); **CCA**, cholangiocarcinoma; **CRC**, colorectal carcinoma; **GC**, gastric or cardia carcinoma; **HCC**, hepatocellular carcinoma.

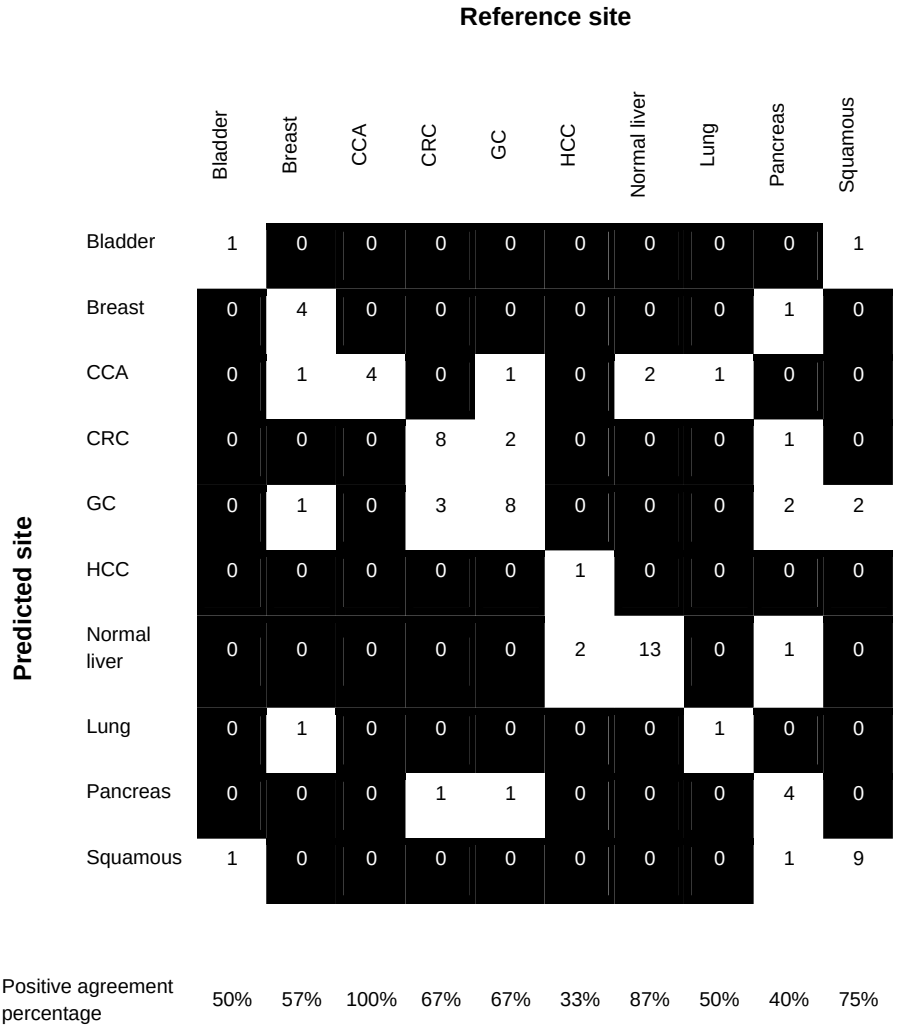
Table 4. High confidence predictions of the CCM+CB classifier.

	Reference site										
	Bladder (5)	Breast (5)	CCA (5)	CRC (5)	GC (5)	HCC (5)	Lung (5)	Pancreas (5)	Squamous cell (5)	Normal liver (10)	Total (55)
High confidence predictions	3	3	2	3	3	5	2	1	4	10	36
Agreement with reference diagnosis	3	3	1	3	2	5	2	1	3	9	32
Positive percentage agreement	100%	100%	50%	100%	67%	100%	100%	100%	75%	90%	89%

The number of high confidence predictions (estimated class probability ≥ 0.6) and the number of high confidence predictions in agreement with the reference diagnosis are listed for the 55 liver core biopsies constituting the independent validation set. The positive percentage agreement was calculated for each individual class and for the overall agreement.
Squamous, squamous cell carcinoma (mixed population); **CCA**, cholangiocarcinoma; **CRC**, colorectal carcinoma; **GC**, gastric or cardia carcinoma; **HCC**, hepatocellular carcinoma.



Supplementary Figure S1,



Supplementary Figure S2.

Chapter 8

Software: Msgl R package

M. Vincent. *msgl: Multinomial sparse group lasso*, 2013. URL `http://cran.r-project.org/web/packages/msgl/index.html`. R package version 0.1.3

Package 'msgl'

May 27, 2013

Type Package

Title High dimensional multiclass classification using sparse group lasso

Version 0.1.3

Date 2013-20-05

Author Martin Vincent

Maintainer Martin Vincent <vincent@math.ku.dk>

Description Sparse group lasso multiclass classification, suitable for high dimensional problems with many classes. Fast algorithm for solving the multinomial sparse group lasso convex optimization problem. This package apply template metaprogramming techniques, therefore -- when compiling the package from source -- a high level of optimization is needed to gain full speed (e.g. for the GCC compiler use -O3). Use of multiple processors for cross validation and subsampling is supported through OpenMP. The Armadillo C++ library is used as the primary linear algebra engine. Armadillo is licensed under the MPL 2.0. The Armadillo C++ library is primarily developed at NICTA (Australia) by Conrad Sanderson, with contributions from around the world. Furthermore the package utilize various Boost libraries, in particular the Tuple library by Jaakko Jarvi and the Random library by Jens Maurer. The Boost libraries are licensed under the Boost Software License.

URL http://arxiv.org/abs/1205.1245 http://arma.sourceforge.net/, http://www.boost.org/

License GPL (>= 2)

LazyLoad yes

Depends R (>= 2.13.0), Matrix, RcppProgress, RcppArmadillo, BH

LinkingTo Rcpp, RcppProgress, RcppArmadillo, BH

Collate 'msgl_multinomial.R'

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-05-27 22:45:46

R topics documented:

`msgl` *Fit a multinomial sparse group lasso regularization path.*

Description

For a classification problem with K classes and p covariates dived into m groups. A sequence of minimizers (one for each lambda given in the `lambda` argument) of

$$\hat{R}(\beta) + \lambda \left((1-\alpha) \sum_{J=1}^{m} \gamma_J \|\beta^{(J)}\|_2 + \alpha \sum_{i=1}^{n} \xi_i |\beta_i| \right)$$

where $\hat{R}$ is the weighted empirical log-likelihood risk of the multinomial regression model. The vector $\beta^{(J)}$ denotes the parameters associated with the J'th group of covariates (default is one covariate per group, hence the default dimension of $\beta^{(J)}$ is K). The group weights $\gamma \in [0, \infty)^m$ and the parameter weights $\xi = (\xi^{(1)}, \ldots, \xi^{(m)}) \in [0, \infty)^n$ with $\xi^{(1)} \in [0, \infty)^{n_1}, \ldots, \xi^{(m)} \in [0, \infty)^{n_m}$.

Usage

```
msgl(x, classes,
  sampleWeights = rep(1/length(classes), length(classes)),
  grouping = NULL, groupWeights = NULL,
  parameterWeights = NULL, alpha = 0.5,
  standardize = TRUE, lambda, return = 1:length(lambda),
  sparse.data = FALSE,
  algorithm.config = sgl.standard.config)
```

Arguments

x	design matrix, matrix of size $N \times p$.
classes	classes, factor of length N.
sampleWeights	sample weights, a vector of length N.
grouping	grouping of covariates, a vector of length p. Each element of the vector specifying the group of the covariate.
groupWeights	the group weights, a vector of length $m+1$ (the number of groups). The first element of the vector is the intercept weight. If groupWeights = NULL default weights will be used. Default weights are 0 for the intercept and $$\sqrt{K \cdot \text{number of covariates in the group}}$$ for all other weights.
parameterWeights	a matrix of size $K \times (p+1)$. The first column of the matrix is the intercept weights. Default weights are is 0 for the intercept weights and 1 for all other weights.
alpha	the α value 0 for group lasso, 1 for lasso, between 0 and 1 gives a sparse group lasso penalty.
standardize	if TRUE the covariates are standardize before fitting the model. The model parameters are returned in the original scale.
lambda	the lambda sequence for the regularization path.
return	the indices of lambda values for which to return a the fitted parameters.
sparse.data	if TRUE x will be treated as sparse, if x is a sparse matrix it will be treated as sparse by default.
algorithm.config	the algorithm configuration to be used.

Value

beta	the fitted parameters – a list of length length(lambda) with each entry a matrix of size $K \times (p+1)$ holding the fitted parameters
loss	the values of the loss function
objective	the values of the objective function (i.e. loss + penalty)
lambda	the lambda values used

Author(s)

Martin Vincent

Examples

```
data(SimData)
x <- sim.data$x
classes <- sim.data$classes
lambda <- msgl.lambda.seq(x, classes, alpha = .5, d = 100L, lambda.min = 0.01)
fit <- msgl(x, classes, alpha = .5, lambda = lambda)
fit$beta[[10]] #model with lambda = lambda[10]
```

`msgl.cv` *Multinomial sparse group lasso cross validation using multiple possessors*

Description

Multinomial sparse group lasso cross validation using multiple possessors

Usage

```
msgl.cv(x, classes, sampleWeights = NULL,
  grouping = NULL, groupWeights = NULL,
  parameterWeights = NULL, alpha = 0.5,
  standardize = TRUE, lambda, fold = 10L,
  cv.indices = list(), sparse.data = FALSE,
  max.threads = 2L, seed = 331L,
  algorithm.config = sgl.standard.config)
```

Arguments

`x`	design matrix, matrix of size $N \times p$.
`classes`	classes, factor of length N.
`sampleWeights`	sample weights, a vector of length N.
`grouping`	grouping of covariates, a vector of length p. Each element of the vector specifying the group of the covariate.
`groupWeights`	the group weights, a vector of length $m+1$ (the number of groups). The first element of the vector is the intercept weight. If `groupWeights = NULL` default weights will be used. Default weights are 0 for the intercept and $\sqrt{K \cdot \text{number of covariates in the group}}$ for all other weights.
`parameterWeights`	a matrix of size $K \times (p+1)$. The first column of the matrix is the intercept weights. Default weights are is 0 for the intercept weights and 1 for all other weights.
`alpha`	the α value 0 for group lasso, 1 for lasso, between 0 and 1 gives a sparse group lasso penalty.
`standardize`	if TRUE the covariates are standardize before fitting the model. The model parameters are returned in the original scale.
`lambda`	the lambda sequence for the regularization path.
`fold`	the fold of the cross validation, an integer larger than 1 and less than $N+1$. Ignored if `cv.indices != NULL`. If `fold`$\leq$`max(table(classes))` then the data will be split into `fold` disjoint subsets keeping the ration of classes approximately equal. Otherwise the data will be split into `fold` disjoint subsets without keeping the ration fixed.

cv.indices	a list of indices of a cross validation splitting. If cv.indices = NULL then a random splitting will be generated using the fold argument.
sparse.data	if TRUE x will be treated as sparse, if x is a sparse matrix it will be treated as sparse by default.
max.threads	the maximal number of threads to be used
seed	the seed used for generating the random cross validation splitting, only used if fold$\leq$max(table(classes)).
algorithm.config	the algorithm configuration to be used.

Value

link	the linear predictors – a list of length length(lambda) one item for each lambda value, with each item a matrix of size $K \times N$ containing the linear predictors.
response	the estimated probabilities - a list of length length(lambda) one item for each lambda value, with each item a matrix of size $K \times N$ containing the probabilities.
classes	the estimated classes - a matrix of size $N \times d$ with $d =$length(lambda).
cv.indices	the cross validation splitting used.
features	average number of features used in the models.
parameters	average number of parameters used in the models.

Author(s)

Martin Vincent

Examples

```
data(SimData)
x <- sim.data$x
classes <- sim.data$classes
lambda <- msgl.lambda.seq(x, classes, alpha = .5, d = 25L, lambda.min = 0.03)
fit.cv <- msgl.cv(x, classes, alpha = .5, lambda = lambda)

# Missclassification count
colSums(fit.cv$classes != classes)
```

---

msgl.lambda.seq *Computes a lambda sequence for the regularization path*

Description

Computes a decreasing lambda sequence of length d. The sequence ranges from a data determined maximal lambda $\lambda_{\max}$ to the user inputed lambda.min.

Usage

```
msgl.lambda.seq(x, classes,
  sampleWeights = rep(1/length(classes), length(classes)),
  grouping = NULL, groupWeights = NULL,
  parameterWeights = NULL, alpha = 0.5, d = 100L,
  standardize = TRUE, lambda.min, sparse.data = FALSE,
  algorithm.config = sgl.standard.config)
```

Arguments

`x`	design matrix, matrix of size $N \times p$.
`classes`	classes, factor of length N.
`sampleWeights`	sample weights, a vector of length N.
`grouping`	grouping of covariates, a vector of length p. Each element of the vector specifying the group of the covariate.
`groupWeights`	the group weights, a vector of length $m+1$ (the number of groups). The first element of the vector is the intercept weight. If `groupWeights = NULL` default weights will be used. Default weights are 0 for the intercept and $\sqrt{K \cdot \text{number of covariates in the group}}$ for all other weights.
`parameterWeights`	a matrix of size $K \times (p+1)$. The first column of the matrix is the intercept weights. Default weights are is 0 for the intercept weights and 1 for all other weights.
`alpha`	the α value 0 for group lasso, 1 for lasso, between 0 and 1 gives a sparse group lasso penalty.
`d`	the length of lambda sequence
`standardize`	if TRUE the covariates are standardize before fitting the model. The model parameters are returned in the original scale.
`lambda.min`	the smallest lambda value in the computed sequence.
`sparse.data`	if TRUE `x` will be treated as sparse, if `x` is a sparse matrix it will be treated as sparse by default.
`algorithm.config`	the algorithm configuration to be used.

Value

a vector of length `d` containing the compute lambda sequence.

Author(s)

Martin Vincent

Examples

```
data(SimData)
x <- sim.data$x
classes <- sim.data$classes
lambda <- msgl.lambda.seq(x, classes, alpha = .5, d = 100L, lambda.min = 0.01)
```

`msgl.subsampling` *Multinomial sparse group lasso generic subsampling procedure*

Description

Support the use of multiple processors.

Usage

```
msgl.subsampling(x, classes,
  sampleWeights = rep(1/length(classes), length(classes)),
  grouping = NULL, groupWeights = NULL,
  parameterWeights = NULL, alpha = 0.5,
  standardize = TRUE, lambda, training, test,
  sparse.data = FALSE, max.threads = 2L,
  algorithm.config = sgl.standard.config)
```

Arguments

`x`	design matrix, matrix of size $N \times p$.
`classes`	classes, factor of length N.
`sampleWeights`	sample weights, a vector of length N.
`grouping`	grouping of covariates, a vector of length p. Each element of the vector specifying the group of the covariate.
`groupWeights`	the group weights, a vector of length $m+1$ (the number of groups). The first element of the vector is the intercept weight. If `groupWeights = NULL` default weights will be used. Default weights are 0 for the intercept and $$\sqrt{K \cdot \text{number of covariates in the group}}$$ for all other weights.
`parameterWeights`	a matrix of size $K \times (p+1)$. The first column of the matrix is the intercept weights. Default weights are is 0 for the intercept weights and 1 for all other weights.
`alpha`	the α value 0 for group lasso, 1 for lasso, between 0 and 1 gives a sparse group lasso penalty.
`standardize`	if TRUE the covariates are standardize before fitting the model. The model parameters are returned in the original scale.

`lambda`	the lambda sequence for the regularization path.
`training`	a list of training samples, each item of the list corresponding to a subsample. Each item in the list must be a vector with the indices of the training samples for the corresponding subsample. The length of the list must equal the length of the `test` list.
`test`	a list of test samples, each item of the list corresponding to a subsample. Each item in the list must be vector with the indices of the test samples for the corresponding subsample. The length of the list must equal the length of the `training` list.
`sparse.data`	if TRUE `x` will be treated as sparse, if `x` is a sparse matrix it will be treated as sparse by default.
`max.threads`	the maximal number of threads to be used
`algorithm.config`	the algorithm configuration to be used.

Value

`link`	the linear predictors – a list of length `length(test)` with each element of the list another list of length `length(lambda)` one item for each lambda value, with each item a matrix of size $K \times N$ containing the linear predictors.
`response`	the estimated probabilities – a list of length `length(test)` with each element of the list another list of length `length(lambda)` one item for each lambda value, with each item a matrix of size $K \times N$ containing the probabilities.
`classes`	the estimated classes – a list of length `length(test)` with each element of the list a matrix of size $N \times d$ with $d =$`length(lambda)`.
`features`	number of features used in the models.
`parameters`	number of parameters used in the models.

Author(s)

Martin Vincent

Examples

```
data(SimData)
x <- sim.data$x
classes <- sim.data$classes
lambda <- msgl.lambda.seq(x, classes, alpha = .5, d = 100L, lambda.min = 0.03)

test <- replicate(5, sample(1:length(classes))[1:20], simplify = FALSE)
train <- lapply(test, function(s) (1:length(classes))[-s])

fit.sub <- msgl.subsampling(x, classes, alpha = .5, lambda = lambda,
 training = train, test = test)

# Missclassification count of second subsample
colSums(fit.sub$classes[[2]] != classes[test[[2]]])
```

predict.msgl *Predict*

Description

Computes the linear predictors, the estimated probabilities and the estimated classes for a new data set.

Usage

```
 ## S3 method for class 'msgl'
predict(object, x, sparse.data = FALSE,
   ...)
```

Arguments

object	an object of class msgl, produced with msgl.
x	a data matrix of size $N_{\text{new}} \times p$.
sparse.data	if TRUE x will be treated as sparse, if x is a sparse matrix it will be treated as sparse by default.
...	ignored.

Value

link	the linear predictors – a list of length length(fit$beta) one item for each model, with each item a matrix of size $K \times N_{\text{new}}$ containing the linear predictors.
response	the estimated probabilities – a list of length length(fit$beta) one item for each model, with each item a matrix of size $K \times N_{\text{new}}$ containing the probabilities.
classes	the estimated classes – a matrix of size $N_{\text{new}} \times d$ with $d =$length(fit$beta).

Author(s)

Martin Vincent

`sgl.algorithm.config` *Create a new algorithm configuration*

Description

With the exception of `verbose` it is not recommended to change any of the default values.

Usage

```
sgl.algorithm.config(tolerance_penalized_main_equation_loop = 1e-10,
  tolerance_penalized_inner_loop_alpha = 1e-04,
  tolerance_penalized_inner_loop_beta = 1,
  tolerance_penalized_middel_loop_alpha = 0.01,
  tolerance_penalized_outer_loop_alpha = 0.01,
  tolerance_penalized_outer_loop_beta = 0,
  tolerance_penalized_outer_loop_gamma = 1e-05,
  use_bound_optimization = TRUE,
  use_stepsize_optimization_in_penalizeed_loop = TRUE,
  stepsize_opt_penalized_initial_t = 1,
  stepsize_opt_penalized_a = 0.1,
  stepsize_opt_penalized_b = 0.1, verbose = FALSE)
```

Arguments

`tolerance_penalized_main_equation_loop`
: tolerance threshold.

`tolerance_penalized_inner_loop_alpha`
: tolerance threshold.

`tolerance_penalized_inner_loop_beta`
: tolerance threshold.

`tolerance_penalized_middel_loop_alpha`
: tolerance threshold.

`tolerance_penalized_outer_loop_alpha`
: tolerance threshold.

`tolerance_penalized_outer_loop_beta`
: tolerance threshold.

`tolerance_penalized_outer_loop_gamma`
: tolerance threshold.

`use_bound_optimization`
: if `TRUE` hessian bound check will be used.

`use_stepsize_optimization_in_penalizeed_loop`
: if `TRUE` step-size optimization will be used.

`stepsize_opt_penalized_initial_t`
: initial step-size.

`stepsize_opt_penalized_a`
: step-size optimization parameter.

stepsize_opt_penalized_b
: step-size optimization parameter.

verbose
: If TRUE some information, regarding the status of the algorithm, will be printed in the R terminal.

Value

A configuration.

Author(s)

Martin Vincent

Examples

```
config.verbose <- sgl.algorithm.config(verbose = TRUE)
```

---

sgl.standard.config *Standard algorithm configuration*

Description

```
sgl.standard.config <- sgl.algorithm.config()
```

Usage

```
sgl.standard.config
```

Format

List of 13 $ tolerance_penalized_main_equation_loop : num 1e-10 $ tolerance_penalized_inner_loop_alpha : num 1e-04 $ tolerance_penalized_inner_loop_beta : num 1 $ tolerance_penalized_middel_loop_alpha : num 0.01 $ tolerance_penalized_outer_loop_alpha : num 0.01 $ tolerance_penalized_outer_loop_beta : num 0 $ tolerance_penalized_outer_loop_gamma : num 1e-05 $ use_bound_optimization : logi TRUE $ use_stepsize_optimization_in_penalizeed_loop: logi TRUE $ stepsize_opt_penalized_initial_t : num 1 $ stepsize_opt_penalized_a : num 0.1 $ stepsize_opt_penalized_b : num 0.1 $ verbose : logi FALSE

Author(s)

Martin Vicnet

---

sim.data *Simulated data set*

Description

The use of this data set is only intended for testing and examples. The data set contains 100 simulated samples grouped into 10 classes. For each sample 400 covariates have been simulated.

Index

Appendix A

Various results

A.1 Quadratic approximations

Consider the linear model setup of section 2.4, hence B is the space of $K \times p$ matrices. We shall, for $\beta \in B$, denote $\operatorname{vec} \beta$ by $\overline{\beta}$, in order to avoid to heavy notation. Consider now the quadratic approximation $Q : B \to \mathbb{R}$

$$Q(\beta) \stackrel{\text{def}}{=} \hat{R}_D(\beta_0) + \nabla \hat{R}_D(\beta_0)^T \left(\overline{\beta} - \overline{\beta}_0\right) + \frac{1}{2} \left(\overline{\beta} - \overline{\beta}_0\right)^T \nabla^2 \hat{R}_D \left(\overline{\beta} - \overline{\beta}_0\right) \tag{A.1}$$

of the empirical risk $\hat{R}_D$ at $\beta_0 \in B$. Often we seek a minimizer of the empirical risk (perhaps regularized) and often such a minimizer is obtained by sequentially optimizing quadratic approximations as (A.1). If the quadratic approximations are not bounded below then such an approach will fail. It is therefore a central question if the quadratic approximations are bounded below or not. For linear models Proposition 6 below provide a sufficient condition in terms of the functions $L_k : \mathbb{R}^K \to \mathbb{R}$ defined by

$$L_k(\eta) \stackrel{\text{def}}{=} L(h(\eta), k)$$

for $k \in \mathcal{S}_K$ and a loss L. Namely, if the gradient of L_k is orthogonal to the kernel of the Hessian of L_k then the quadratic approximation (A.1) is bounded below.

A.1.1 The gradient and the Hessian

The gradient and the Hessian of the empirical risk of linear models has a specific form as can be seen in Proposition 5 below. For any loss L and any linear model h we have that;

Proposition 5. *The equalities*

$$\nabla \hat{R}_D(\beta) = \frac{1}{N} \sum_{i=1}^{N} x_i \otimes \nabla L_{y_i}(\beta x_i)$$

and

$$\nabla^2 \hat{R}_D(\beta) = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T \otimes \nabla^2 L_{y_i}(\beta x_i)$$

holds.

Proof. For the first equality we need to show that, for $k \in \mathcal{S}_K$, the gradient of $R_k(\beta) \stackrel{\text{def}}{=} L_k(\beta x)$ is

$$x^T \otimes \nabla L_k(\beta x).$$

Define $\eta_x : B \to \mathbb{R}^K$ by $\eta_x(\beta) \stackrel{\text{def}}{=} \beta x$, then $R_k(\beta) = L_k \circ \eta_x$. The differential of R_k is

$$\begin{aligned} \mathrm{d}R_k &= \mathrm{d}L_k \circ \eta_x) \\ &= \mathrm{d}L_k \, \mathrm{d}\beta x. \end{aligned}$$

This implies that

$$\mathrm{d}R_k = x^T \otimes \mathrm{d}L_k \, \mathrm{d}\operatorname{vec}\beta.$$

The first equality now follows by the first identification theorem, see Magnus and Neudecker [10].

The differential of the transposed of the gradient $\nabla R_k = x \otimes \nabla L_k(\beta x)$ is

$$\left(x \otimes \mathrm{d}(\nabla L_k^T)(\beta x)\right) \mathrm{d}\beta x.$$

This implies that

$$\mathrm{d}\operatorname{vec}\nabla R_k^T = x^T \otimes \left(x \otimes \nabla^2 L_k(\beta x)\right) \mathrm{d}\operatorname{vec}\beta.$$

The second equality now follows by the first identification theorem. □

A.1.2 The multinomial regression model

For the multinomial regression model the gradient of L_k is

$$\nabla L_k(\eta) = h(\eta) - e_k,$$

and the Hessian is

$$\nabla^2 L_k(\eta) = \operatorname{diag}(h(\eta)) - h(\eta)^T h(\eta).$$

The kernel of the Hessian is seen to be the subspace generated by the single vector $v = e_1 + \cdots + e_K$. Since $(e_l - \tilde{p})^T v = 0$ for all $l = 1, \ldots, K$ the conditions in Proposition 6 are fulfilled; hence, the quadratic approximations of the multinomial log likelihood are bounded below. It is desirable that the quadratic approximations are bounded below as this ensures that a minimizer, of the quadratic approximation, exists.

Proposition 6. *If the gradient $\nabla L_k(\eta)$ is orthogonal to the kernel of the Hessian $\nabla^2 L_k(\eta)$ for all $\eta \in \mathbb{R}^K$ and $k \in \mathbb{S}_K$ then Q is bounded below.*

Proof. Define for each $i = 1, \ldots, N$ the quadratic function $Q_i : \mathbb{R}^{K \times p} \to \mathbb{R}$ by

$$Q_i(\beta) \stackrel{\text{def}}{=} \left[x_i \otimes \nabla L_{y_i}(\beta_0 x_i)\right]^T \left(\overline{\beta} - \overline{\beta}_0\right) + \frac{1}{2} \left(\overline{\beta} - \overline{\beta}_0\right)^T \left[x_i x_i^T \otimes \nabla^2 L_{y_i}(\beta_0 x_i)\right] \left(\overline{\beta} - \overline{\beta}_0\right).$$

Then $Q(\beta) = \hat{R}_D(\beta_0) + \sum_{i=1}^N Q_i(\beta)$ and since, by lemma 6, each of the Q_i quadratic functions are bounded below then we are done. □

Lemma 6. *Assume that the gradient $\nabla L_k(\eta)$ is orthogonal to the kernel of the Hessian $\nabla^2 L_k(\eta)$. Then the quadratic function*

$$\left[x \otimes \nabla L_k(\eta)\right]^T \left(\overline{\beta} - \overline{\beta}_0\right) + \frac{1}{2} \left(\overline{\beta} - \overline{\beta}_0\right)^T \left[x x^T \otimes \nabla^2 L_k(\eta)(\beta_0 x)\right] \left(\overline{\beta} - \overline{\beta}_0\right)$$

is bounded below.

Proof. Let $V = \ker \nabla^2 L_k(\eta)$; hence, V is a subspace of $\mathbb{R}^K$. Denote by $v_1, \ldots, v_m$ a basis for V, and extend this to a basis $v_1, \ldots, v_K$ for $\mathbb{R}^K$. Choose a basis $w_1, \ldots, w_p$ for $\mathbb{R}^p$. The collection $w_j \otimes v_k$ for $j = 1, \ldots p$, $k = 1, \ldots, K$ is then a basis for the tensor product $\mathbb{R}^K \otimes \mathbb{R}^p \simeq \mathbb{R}^{K \times p}$. If

$$\left(xx^T \otimes \nabla^2 L_k(\eta)\right) w_j \otimes v_k = 0$$

then $x_i^T w_j = 0$ or $v_k \in V$. It follows that, in this case,

$$\left(x_i \otimes \nabla L_k(\eta)\right)^T w_j \otimes v_k = 0.$$

The statement now follows by Lemma 7. □

Lemma 7. *Let $q \in \mathbb{R}^n$ and let A be any symmetric positive semi-definite $n \times n$ matrix. If q is orthogonal to the kernel of A, then the quadratic function $Q : \mathbb{R}^n \to \mathbb{R}$ defined by*

$$Q(z) = q^T z + z^T A z$$

is bounded below.

Proof. Let V denote the kernel of A, and assume that q is orthogonal to V. Then for any vector $v \in \mathbb{R}^n$

$$Q(v) = Q(Pv)$$

where P is the projection onto the orthogonal complement of V. Q is bounded below on the orthogonal complement $V^\perp$ of V, since $P^T A P$ is positive definite on $V^\perp$. □

A.2 Identifiability and parameter interpretation

For a parametric model $\mathfrak{p} : B \times \mathbb{R}^p \to \Delta^K$ the set *of parameterizable models* is

$$\mathcal{P} \stackrel{\text{def}}{=} \{\mathfrak{q} : \mathbb{R}^p \to \Delta^K \mid \text{there exists } \beta \in B \text{ such that } \mathfrak{q} = \mathfrak{p}(\beta)\}.$$

The set $\mathcal{P}$ is a subset of the set of all conditional distributions on Y given X. It is the image of the function $B \xrightarrow{\phi} \mathcal{P}$, defined by mapping β to the function $x \to \mathfrak{p}(\beta)(x)$. The set $E_\mathfrak{q} \stackrel{\text{def}}{=} \phi^{-1}(\mathfrak{q}) \subseteq B$ is called the *fiber* over $\mathfrak{q} \in \mathcal{P}$. The fibers partitions the parameter set B into disjoint sets, i.e.

$$B = \coprod_{\mathfrak{q} \in \mathcal{P}} E_\mathfrak{q}$$

where II is denotes disjoint union; an ordinary union in which the sets are disjoint.

The set $\mathcal{P}$ is the set of all, by the model, reachable classifiers. The question of *identifiability* is the question of the structure of the fibers of the model. Each fiber corresponds to a unique classifier, hence the model can be used to identity parameters modulo the structure of the fibers.

When dealing with classification we are sometimes only interested in using the model to make predictions, in such cases identifiability is not an issue we need to consider. If we wish to interpret the estimated parameters of the model then we need to have in mind the structure of the fibers.

The strongest form of identifiability is when each parameter corresponds to a unique classifier. That is;

Definition 19 (Strong identifiability)**.** *A model is said to be strong identifiable if the fibers of the model are singletons.*

Strong identifiability is the usual notion of identifiability. However strong identifiability is too restrictive for the models and estimators we will study.

In the linear case where B can be structured as the space of $K \times p$ matrices, we can define linear identifiability. We will say that;

Definition 20 (Linear identifiability)**.** *A model is linear identifiable if the fibers of the model are*

$$\beta_0 + \mathbb{1}_K \otimes \mathbb{R}^p$$

for $\beta_0 \in B$ and where $\mathbb{1}_K$ is the K dimensional vector of all ones [1]*.*

A.2.1 Identifiability of regular linear models

Let h be a regular linear model. The fibers over $\beta_0 \in B$ is

$$\{\beta \in B \mid \mathfrak{p}(\beta) = \mathfrak{p}(\beta_0)\} = \{\beta \in B \mid \beta x \in h^{-1}(\beta_0 x) \text{ for all } x \in \mathbb{R}^p\}.$$

It follows by Lemma 8 that; the fiber over $\beta_0 \in B$ is the set of all $\beta \in B$ for which there exist a function $c : \mathbb{R}^p \to \mathbb{R}_+$ such that

$$c(x)g(\eta_k(x)) = g(\tilde{\eta}_k(x)) \text{ for all } x \in \mathbb{R}^p$$

for all $k \in \mathcal{S}_K$ and where $\eta(x) = \beta x$ and $\tilde{\eta}(x) = \beta_0 x$.

Lemma 8. *Let $\xi \in \Delta^K$ then for a regular linear model h*

$$h^{-1}(\xi) = \{\eta \in \mathbb{R}^K \mid \textit{there exist } c > 0 \textit{ such that } g(\tilde{\eta}_k) = cg(\eta_k) \textit{ for all } k \in \mathcal{S}_K\}$$

where $\tilde{\eta} \in h^{-1}(\xi)$.

Proof. It follows directly form Lemma 2 that the inclusion $\supseteq$ holds. To see that the other inclusion holds let $\eta \in h^{-1}(\xi)$ and $c = g(\tilde{\eta}_1)/g(\eta_1)$. Since $h(\eta) = h(\tilde{\eta}) = \xi$ we have that

$$\frac{g(\tilde{\eta}_1)}{\sum_{k=1}^K g(\tilde{\eta}_k)} = \xi_1 = \frac{g(\eta_1)}{\sum_{k=1}^K g(\eta_k)}$$

which implies that

$$\frac{\sum_{k=1}^K g(\eta_k)}{\sum_{k=1}^K g(\tilde{\eta}_k)} = \frac{g(\eta_1)}{g(\tilde{\eta}_1)} = \frac{1}{c}. \tag{A.2}$$

Now for $l \in \mathcal{S}_K$ we have that

$$\frac{g(\tilde{\eta}_l)}{\sum_{k=1}^K g(\tilde{\eta}_k)} = \xi_l = \frac{g(\eta_l)}{\sum_{k=1}^K g(\eta_k)}$$

which by (A.2) implies that

$$g(\tilde{\eta}_l) = cg(\eta_l).$$

□

[1] $\beta_0 + \mathbb{1}_K \otimes \mathbb{R}^p$ is short notation for the set $\{\beta_0 + \mathbb{1}_K \otimes v \mid v \in \mathbb{R}^p\}$

A.2.2 Identifiability of the (symmetric) multinomial regression model

Theorem 5. *The (symmetric) multinomial regression model is linear identifiable.*

Proof. By Lemma 8 we have that the fiber over $\mathfrak{p}(\tilde{\beta})$ is the union of

$$\begin{aligned} A(c) &= \{\beta \in \mathbb{R}^{K\times p} \mid \exp(\beta x) = \exp(\tilde{\beta}x + c(x)\mathbb{1}) \text{ for all } x \in \mathbb{R}^p\} \\ &= \{\beta \in \mathbb{R}^{K\times p} \mid (\beta - \tilde{\beta})x = c(x)\mathbb{1} \text{ for all } x \in \mathbb{R}^p\} \end{aligned}$$

over all functions $c : \mathbb{R}^p \to \mathbb{R}$. Since $A(c) = \emptyset$ if c is non-linear, it follows that the fiber over $\mathfrak{p}(\tilde{\beta})$ is

$$\bigcup_{\{c:\mathbb{R}^p\to\mathbb{R} \mid c \text{ is linear}\}} A(c).$$

Which implies that the fiber is

$$\{\beta \in \mathbb{R}^{K\times p} \mid \text{there exist } c \in \mathbb{R}^p \text{ such that } (\beta - \tilde{\beta})x = c^T x\mathbb{1}\}.$$

This can be rewritten as

$$\{\beta \in \mathbb{R}^{K\times p} \mid \text{there exist } c \in \mathbb{R}^p \text{ such that } \beta = \tilde{\beta} + \mathbb{1} \otimes c^T\}.$$

□

Appendix B

Results from convex analysis

In this appendix some results from convex analysis are collected, details can be found in Urruty and Lemaréchal [18].

B.0.3 Set operations

We will use the following convex preserving set operations;

- For a convex set C scalar multiplication by $\lambda \in \mathbb{R}$ is the convex set

$$\lambda C \stackrel{\text{def}}{=} \{\lambda s \mid s \in C\}. \tag{B.1}$$

- For a affine map A the image $A(C)$ of C is the convex set

$$AC \stackrel{\text{def}}{=} A(C). \tag{B.2}$$

- For convex sets C_1 and C_2 the sum is the convex set

$$C_1 + C_2 \stackrel{\text{def}}{=} \{t + s \mid t \in C_1, s \in C_2\}. \tag{B.3}$$

B.0.4 Sublinear function

A function $\sigma : R^n \to \mathbb{R}$ is said to be (finite) *sublinear* if

$$\sigma(t_1 x_1 + t_2 x_2) \leq t_1 \sigma(x_1) + t_2 \sigma(x_2) \tag{B.4}$$

for all $x_1, x_2 \in \mathbb{R}^n$ and all $t_1, t_2 > 0$. A function is sublinear if and only if it is convex and positively homogeneous. Examples of sublinear functions are norms, semi norms and linear functions.

B.0.5 Support function

For a compact convex set $C \subseteq \mathbb{R}^n$ the *support function* of C is the function $\sigma_C : \mathbb{R}^n \to \mathbb{R}$ defined by

$$\sigma_C(x) \stackrel{\text{def}}{=} \sup\{s^T x \mid s \in C\}. \tag{B.5}$$

We note that following properties of support functions can be shown;

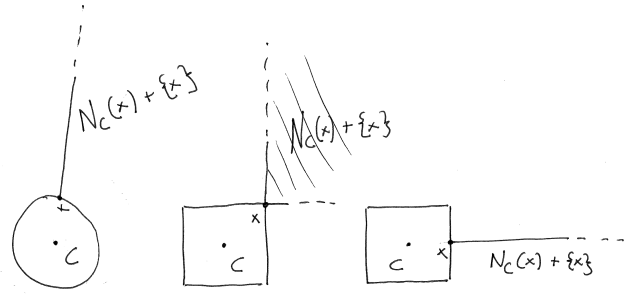


Figure B.1: Examples of normal cones

- Sublinear functions is in a bijective correspondence with support functions of compact convex sets.
- For a symmetric matrix A and a support function σ_C it follows by the definition that $\sigma_C(Ax) = \sigma_{AC}(x)$.
- For convex sets C_1 and C_2 $\sigma_{C_1}(x) + \sigma_{C_2}(x) = \sigma_{C_1+C_2}(x)$.

B.0.6 Normal cone

For a convex set C the *normal cone* to C at x is

$$N_C(x) \stackrel{\text{def}}{=} \{s \in \mathbb{R}^n \mid s^T(y-x) \leq 0 \text{ for all } y \in C\}. \tag{B.6}$$

The normal cone is a *cone*.

B.0.7 Projection onto a convex set

Let $C \subseteq \mathbb{R}^p$ be a compact convex set, the projection of $x \in \mathbb{R}^p$ onto C is the element in C with smallest distance to x, i.e. a solution to

$$\underset{y \in C}{\text{minimize}} \, \|y - x\|_2^2 \,. \tag{B.7}$$

It can be shown that (B.7) always have a unique solution. We may therefore define the projection onto C as the function $P_C : \mathbb{R}^p \to \mathbb{R}^p$ given by

$$P_C(x) \stackrel{\text{def}}{=} \underset{y \in C}{\arg\min} \, \|y - x\|_2^2 \,. \tag{B.8}$$

The following properties can be shown

- For $x \in \mathbb{R}^p$
$$t = P_C(x) \iff (x-t)^T(y-t) \leq 0 \quad \text{for all } y \in C. \tag{B.9}$$
- For $x \in C$ and $t \in \mathbb{R}^p$ it holds that
$$t \in N_C(x) \iff x = P_C(x+t). \tag{B.10}$$

Appendix C

Data examples

- **Primary Cancers** This data set is a subset of the data used in Perell et al. [12] (Chapter 7). The data set consist of miRNA expression measurements of leaser dissected primary cancers.
- **Childhood Leukemia** Gene expression in acute lymphoblastic leukemia cells after treatment with methotrexate and mercaptopurine given alone or in combination, Cheok et al. [4]. Genes not present in at least one sample were removed.
- **Brain tumor** The data set consist of mRNA measurements of 4 different tumors of the central nervous system and normal cerebellum, Pomeroy et al. [13].
- **Amazon reviews** The Amazon review data set consists of 10k textual features (including lexical, syntactic, idiosyncratic and content features) extracted from 1500 customer reviews from the Amazon Commerce Website. The reviews were collected among the reviews from 50 authors with 50 reviews per author. The primary classification task is to identify the author based on the textual features. The data and feature set were presented in [9] and can be found in the UCI machine learning repository [6].

Appendix D

Article: Efficient identification of metastases

R. Søkilde, M. Vincent, A. K. Møller, A. Hansen, P. E. Høiby, T. Blondal, B. S. Nielsen, G. Daugaard, S. Møller, and T. Litman. Efficient identification of metastases by their microrna profile

Efficient identification of metastases by their microRNA profile

Rolf Søkilde[1,4+], Martin Vincent[1,5], Anne Kirstine Hundahl Møller[2], Alastair Hansen[3], Poul Erik Høiby[1], Thorarinn Blondal[1], Boye Schnack Nielsen[1,6], Gedske Daugaard[2], Søren Møller[1,7], and Thomas Litman[1,8+]

1 Department of Biomarker Discovery, Exiqon A/S, DK-2950 Vedbæk.

2 Department of Oncology, Copenhagen University Hospital Rigshospitalet, DK-2100 Copenhagen Ø.

3 Department of Pathology, Herlev University Hospital, DK-2730 Herlev, Denmark.

4 University of Lund, Department of Oncology, Sweden.

5 University of Copenhagen, Department of Mathematical Sciences, Copenhagen, Denmark

6 Bioneer, Hørsholm, Denmark

7 Novozymes A/S, Bagsvaerd, Denmark

8 University of Copenhagen, Department of International Health, Immunology and Microbiology, Copenhagen, Denmark

+Corresponding author. Email: rolf.soekilde@med.lu.se, +45 25 32 68 67, Barngatan 2B, 22185 Lund, Sweden

Running head: microRNA based identification of metastases

Support: The Danish National Advanced Technology Foundation supported this work with grant no. 048-2008-1

Highlights:

Affiliations: At the time of this study, RS, MV, TB, PEH, BSN, SM, and TL were employed at Exiqon A/S, which has a direct financial interest in the subject matter discussed.

Abstract

Carcinomas of unknown primary origin constitute 3-5% of all newly diagnosed metastatic cancers, of which the primary source is difficult to classify with current histological methods. Effective cancer treatment depends on early and accurate identification of the tumor, which is why patients with metastases of unknown origin have poor prognosis and short survival. Because microRNA expression is highly tissue specific, the microRNA profile of a metastasis may be used to identify its origin. As a first step to realize this goal, we evaluated the potential of microRNA profiling for identification of both the primary tumor and of its metastases.

208 formalin-fixed paraffin-embedded samples representing 15 different histologies were profiled on an LNA-enhanced microarray platform, which allows for highly sensitive and specific detection of microRNA. Based on these data, we developed and cross-validated a novel classification algorithm, LASSO (least absolute shrinkage and selection operator), which had an overall accuracy of 85%. When the classifier was applied on an independent test set of 48 metastases, the primary site was correctly identified in 42 cases (88% accuracy).

Our findings suggest, that microRNA expression profiling on paraffin tissue can efficiently predict the primary origin of a tumor, and may provide pathologists with a molecular tool that can improve their capability to correctly identify the origin of hitherto unidentifiable metastatic tumors, and eventually, enable tailored therapy.

Keywords:

Non-coding RNA; Expression; LASSO; FFPE; Feature selection

Abbreviations

ANN, artificial neural network

CUP, carcinoma of unknown primary origin

FFPE, formalin-fixed paraffin-embedded

H&E, Hematoxylin-eosin

KNN, K nearest neighbors

LASSO, least absolute shrinkage and selection operator

LDA, linear discriminant analysis

LNA, locked nucleic acid

miRNA, microRNA

SVM, support vector machine

Introduction

Carcinoma of unknown primary site (CUP) represents a heterogeneous group of metastatic malignancies for which no primary site of the tumor can be identified following a thorough medical history, careful clinical examination and extensive diagnostic work-up. CUP accounts for approximately 5 % of all cancer diagnoses and represents the seventh most frequent type of cancer. It is characterized by early dissemination, aggressive biology, uncommon metastatic sites, resistance to therapy and usually, a poor prognosis[1]. Even with an extensive diagnostic work-up using advanced immunohistochemical and imaging techniques, the frequency of detecting the primary tumor site remains low. In less than 30 % of CUP patients a primary tumor site is identified ante mortem. Post mortem examinations reveal a putative primary tumor site in 60-80 % of CUP patients, most often in the lung (27 %), pancreas (24 %) or in the hepatobiliary tree (8 %)[2]. Failure to identify the primary tumor site may negatively influence patient management, as tailored chemotherapeutic regimens and targeted agents are being developed for a number of solid tumors.

Cancer classification based on gene expression profiling by DNA microarrays was demonstrated already in 1999 for leukemia by Golub et al.[3], and subsequently, has been extended to include categorization of solid tumors and their metastases[4–11]. Therefore, gene expression profiling could be an important diagnostic tool in CUP patients by predicting the primary tumor site and thus, enabling tailored organ-specific therapy that hopefully can be translated into improved survival.

MicroRNAs (miRNAs) constitute a recently discovered class of tissue specific, small, non-coding RNAs, which regulate the expression of genes involved in many biological processes, including development, differentiation, apoptosis and carcinogenesis[12,13]. Several studies have shown that miRNAs are promising molecular biomarkers for classification of cancer[14–17]. Besides their tissue specificity, a major advantage of miRNAs is their short size, which renders them more stable in formalin-fixed, paraffin-embedded (FFPE) material compared to mRNA[18,19].

In the present study, we developed a miRNA classifier and evaluated its potential to predict the origin of the primary tumor in cancer patients. By applying a microarray platform based on locked

nucleic acid (LNA) modified detection probes [20], which enable highly sensitive and specific detection of miRNAs, we identified tissue specific microRNA signatures for 35 tumors and histologies, of which 15 were included in a novel multi-class classification algorithm that can predict the site of tumor origin with high accuracy.

Results

Sample selection

To obtain a comprehensive data set for constructing the microarray tumor database, we initially profiled 408 tissue samples covering 35 different histologies (data not shown). Selection of the tumor classes for the final classifier was based on the results from autopsy studies in CUP patients: More than 75% of all CUP cases are adenocarcinomas and poorly differentiated carcinomas, of which the most common primary tumor sites identified at autopsy are pancreas (25%), lung (20%), stomach, colorectal, and hepatobiliary tract (8-12% each), and kidney (5%). Squamous cell carcinomas account for 5-10 % where the primary site most often originates from head and neck cancers, while melanoma represent 4% of all CUP cases[21]. However, these relative frequencies should be interpreted with caution, as the epidemiology of CUP is changing due to both improved imaging technology and lifestyle habits, and therefore, different studies report very dissimilar frequencies of identified primary sites[2].

Based on the above considerations, our final classifier includes profiles from 15 known cancer classes with 12 carcinomas as well as melanoma, germ cell tumors, and lymphoma, as the latter can be difficult to distinguish from poorly differentiated carcinoma. The classifier was developed on FFPE material as it is readily available, miRNAs are stable in FFPE blocks, and straightforward to extract [22] . **Table 1** lists the training set of 208 FFPE samples (199 primary tumors and 9 metastases) representing 15 known cancer classes and their histologies (columns 1 and 2). A detailed summary of the patient demographic data can be found in **Supplementary Table S1**.

[Table 1]

Tissue specific miRNA expression

The distribution of tissue specific miRNAs (i.e. those miRNAs that were preferentially expressed in samples originating from one tissue compared to all other tissues) is summarized in the heat-map below (**Fig. 1).**

[Figure 1]

From the heat-map it is evident that some histologies are easy to distinguish from the rest due to a strong and homogeneous tissue specific miRNA signature (adrenal, lymphoma, germ cell, prostate, GIST, and melanoma), while other tissue origins are more difficult to classify accurately, mainly because of heterogeneity within the group (ovary, lung), or because of high similarity to related tissue types (colorectal and EG-junction).

Feature selection

Because selection of the candidate biomarkers is crucial for the performance of the classifier, we took several different approaches to identify the best possible tissue-specific markers. The first – and simplest – approach was to run "one-against-one" and "one-against-all" comparisons for each cancer class/tissue, identifying differentially expressed miRNAs by t-tests. However, running multiple two-sample t-tests may result in an increased risk of committing a Type I error (false positive), which is why we also applied ANOVA (analysis of variance) to compare all 15 means (of the different cancer classes) in one test. Yet, because filtering based methods, such as t-test and ANOVA, do not provide a cross validation option for optimizing the set of discriminatory features, we decided for an embedded approach, namely the LASSO method, which integrates feature selection within the classifier construction. With this method 132 miRNAs with high tissue discriminatory potential were identified; these are listed in **Supplementary Table S2,** which is a data matrix showing each feature's LASSO model coefficient for the particular tissue of interest.

Finally, we made a literature search for tissue-specific miRNAs, and compared these to our top candidate discriminatory miRNAs. There was, not surprisingly, a high degree of overlap between the miRNAs identified in our study and those reported previously as having high predictive ability for cancer classification[15–17]. The overlapping miRNAs are also indicated in **Supplementary Table S2**.

Classifier performance

Many different algorithms are available for multiclass classification and feature selection, such as KNN[23], genetic algorithm (GA)[10], linear discriminant analysis (LDA)[24], support vector machine (SVM)[11], recursive feature elimination[4], nearest shrunken centroids[15], decision trees[16,25], and artificial neural networks[26,27].

One of the main objectives of this study was to combine feature selection and multiclass classification into one process, which should be able to integrate identification of highly informative features useful for classification with cross validation of the results. This dual function is not offered by most other commonly used algorithms, which is why we decided to remodel the LASSO algorithm for this purpose[28]. Specifically, we wished to optimize the model to obtain as high sensitivity (and accuracy) on all 15 tumor classes as possible. This is illustrated in **Figure 2**, which shows the performance of the LASSO classifier as a function of the regularization parameter.

[Figure 2]

The results of the 5-fold cross validation of the LASSO classifier are illustrated in **Table 2**, which is a confusion matrix, showing the number of correct classifications along the diagonal. The correct tissue of origin was predicted in the vast majority of cases (176 of 208 samples tested) with an overall accuracy of 85%. Typically, the false-positive calls were due to similarities in histology causing cross-reactivity; for example, three gastro-esophageal (EG-junction) samples were wrongly predicted as colorectal. We were not able to separate stomach cancers from esophageal adenocarcinomas based on their miRNA profile, why we decided to pool these two, rather similar histologies, which is consistent with other, recent miRNA profiling studies[16,17].

[Table 2]

Validation on metastatic samples

Except for melanoma, the LASSO classifier was built on primary tumors. Therefore, it was important to validate its performance in an independent test set consisting of metastases (n=48) from different sites, including liver, lymph nodes and peritoneum, to ensure that over-fitting to the original training data was not an issue. The results of the validation are summarized in **Table 3**. During the optimization of the classifier, we discovered that even though the validation samples all contained less than 25% normal surrounding tissue, the signal from especially the liver, classified most metastases to the liver as cholangiocarcinoma. Therefore, it was necessary to add the rule to the classifier that the site of metastasis cannot be classified as the primary tumor (i.e. metastasis to the liver is excluded from being identified as a primary liver tumor). The main prediction of the LASSO classifier was correct in 33/48 cases, or 42/48 cases (88% accuracy), considering both the first (33 cases) and the second (9 cases) classification attempt. Thus, the classification of the independent test set consisting of metastatic samples only showed that the performance of the LASSO classifier was comparable to the estimates from the 5-fold cross validation. The same trend of misclassification of the digestive system is seen for the metastatic samples, as for the primary tumors.

[Table 3]

Discussion

CUP represents a well-recognized and important clinical problem, because optimal treatment selection depends on a correct identification of the site of origin, which is per definition occult in a patient presenting with CUP. Therefore, many attempts have been made to improve diagnostic pathology workup of CUP, ranging from purely immunohistochemical schemes for sub-typing the tumor[29,30], over combined

classification approaches[27], to proteomic analysis [31] and machine learning algorithms based on large-scale mRNA microarray profiling [4,5,9,11,32–34] or on RT-PCR data [10,24,35,36]. Recently, miRNAs, which are characterized by their highly tissue specific expression, have also been reported as useful for classification of tumor types[14–17].

In this study, we have applied an LNA-enhanced microarray platform to generate miRNA expression profiles from 208 FFPE samples representing 15 different tumor histologies. The miRNA data were used to successfully develop and validate a novel classification scheme, based on the LASSO algorithm, which integrates feature selection within the classifier construction[28]. The accuracy of the LASSO algorithm was 85% when assessed by 5-fold cross validation on the initial training set, and 88% when applied on an independent test set of 48 known metastases (considering both the first and second classification attempt). Thus, the current approach has approximately the same sensitivity as other multi-class classification methods[10,16]. Validation on more metastases, representing more histological classes and metastatic locations would be valuable, but is limited by the scarcity of metastatic samples.
Where the LASSO method shows its strength, is its approximately equal sensitivity to all the classes in the classifier. Other methods may suffer from very poor performance on a few classes; for example, the combined tree and KNN based miRNA classifier reported by Rosenfeld et al. has 0 (zero) sensitivity to bladder cancer[16], while the LASSO algorithm detects this histology with a mean sensitivity of 95%.

Identifying the algorithm that is best suited for clinical use is an ongoing and controversial discussion. It has been argued that “black box” machine learning classifiers, such as SVM and ANN, are not as transparent as e.g. decision trees for practical use by pathologists[16,27]. However, in spite of their intuitive and visual appeal, decision trees are not without limitations: if they become over-complex they do not generalize the data well, and there is no backtracking option, meaning that a local (erroneous) optimal solution will prevent one from reaching the global optimal solution, e.g. the correct classification will be missed once a wrong path is followed down a branch [37]. In this respect, it is interesting that the binary decision tree originally proposed by Rosenfeld et al for miRNA classification of cancer tissue[16], has undergone substantial structural changes in the follow-up study by Rosenwald et al[17], resulting in a more complex tree (with 12 branch points for some class labels) and more than half of the 48 miRNAs reported in

the original study replaced by other, tissue-specific miRNAs. This adjustment of tree structure probably reflects both the altered tissue selection and the optimized selection of features.

A recent paper by Centeno et al. suggests a hybrid, decision tree model, which incorporates both IHC and expression data for optimal separation of four types of carcinoma[27]. However, one should bear in mind that interpretation of IHC staining is subjective and therefore, it can be difficult to determine a positive from a negative (as Gown's fourth law of immunohistochemistry laconically states: "All that turns brown is not positive"[38]). A recent meta-analysis performed by Anderson and Weiss showed that IHC only provides correct tissue identification in 65.6% of metastatic cancers[39], which underscores the need for improved identification of the primary site of metastatic cancers.

We believe that the LASSO algorithm offers the best of both worlds – that is: the performance of the complex machine learning algorithm together with the intuitive understanding of the simpler classifiers – as it is very powerful and easy to train, the model complexity (number and type of features) can be easily controlled, over-fitting is restricted by a penalty term, and data interpretation is simple: the read-out is the likelihood of a correct classification. Other conventional methods, such as LDA and KNN, resulted in less accurate classification (data not shown) of this data compared to LASSO.

We were able to identify the origin of metastatic tumors by their miRNA profiles, and this is consistent with the paradigm that the genetic makeup of a primary tumor is retained in the distant metastases[8,16,24]. Several of the identified tissue-specific miRNAs are involved in differentiation, so if the miRNA signature is retained in the metastases, it should be possible to identify its tissue of origin, unless the cancer is so dedifferentiated, that all molecular marks of its primary origin are lost. This brings up the question if a "real" CUP represents an entity of its own, with a "CUP-specific" rather than a primary tissue specific molecular signature[4,40–42].

Some tissues are inherently difficult to classify correctly, for example pancreas cancer, which is often poorly differentiated or dedifferentiated, and lung cancer with many possible histologies. In our validation study, the classifier was able to correctly label pancreas as the primary site in 3 out of 5 cases; this is an improvement compared to what could be achieved in the commercial CupPrint follow-up study[34],

where none of the three pancreas cancers could be identified. Additionally, Park et al. [43] found that with IHC markers the sensitivity towards the combined group of pancreas cancer and cholangiocarcinoma was quite low (28%).

We discovered that a major limitation to this type of study is the identification of the superimposed host tissue (the site of the metastasis) signature - typically liver or lymph node - rather than the metastasis signature, in particular, when the amount of host tissue is large compared to the metastasis. Therefore, when optimizing the classifier, we had to make the assumption that a metastasis to the liver cannot be primary liver cancer, and that a lymph node metastasis is not a lymphoma. A mixture model approach as suggested by Ghosh [44] and by Clarke et al.[45], trying to subtract the host tissue signature from the metastasis signature appeared problematic, mainly due to the difficulty in estimating the ratio of tumor to normal tissue. A likely solution to the problem of contaminating surrounding tissue is to apply laser capture micro-dissection[46], as suggested by Chen et al for miRNA analysis in intrahepatic cholangiocarcinoma[47].

In conclusion, our study suggests that microRNA expression profiling on FFPE tissue, followed by an efficient multi-class classification algorithm – in this case LASSO - can efficiently predict the primary origin of a tumor. Thus, it may provide pathologists with an adjunct molecular tool that either alone or in combination with other relevant biomarkers, such as mRNA and proteins e.g. automated IHC), can improve their capability to correctly identify the origin of metastatic tumors, and eventually, advance and expedite rational, specific therapy of patients with metastatic disease.

Materials and Methods

Tumor samples

More than 400 FFPE tumor (both primary and metastases) and normal adjacent tissue samples were procured from National Disease Research Interchange (NDRI, Philadelphia, PA), Cytomyx (Lexington, MA),

Proteogenex (Culver City, CA), and our in-house tissue bank. Every sample was obtained with a copy of its anonymized pathological report, and both the pathology information and an H&E section of each preparation was reviewed by a pathologist (AH) to ascertain the diagnosis, origin, and tumor percentage of the sample. Inclusion criteria for subsequent RNA extraction and miRNA expression analysis were: >0.5 mm^2 tumor size, <25% normal adjacent tissue < 20% necrosis or hemorrhage, and confirmed histology. In the pilot phase of the project, we collected 408 samples from 35 different tumor histologies to cover a broad selection of solid tumors, whereas for the classifier, we narrowed down the list of included tissues to 15 (represented by 208 FFPE samples), to represent only the clinically most relevant histologies for identification of tumors of unknown origin (**Table 1**). All demographic metadata were deposited in a database, and are available in **Supplementary Table S1**. For validation of the classifier, an independent set of 48 metastases with known origin was collected from NDRI, and our in-house tissue-bank.

RNA isolation

Total RNA was extracted from 20µm FFPE sections with the High Pure miRNA Isolation Kit (Roche Applied Science, Mannheim, Germany) according to the manufacturer's instructions. After elution in 40µl RNase free water, the RNA concentration (A260 nm) and purity (A260/280 and A260/230 ratios) were assessed with a Nanodrop ND-1000 spectrophotometer (Thermo Scientific, Wilmington DE). The RNA was stored at -80 C until further analysis.

Microarray profiling

For microarray analysis, we applied a common reference design, where the reference sample contains a mixture of total RNA representing all tissue types in the study. This allows for both one- and two-channel data analysis, as described in detail by Søkilde et al.[48]. In the current study, we applied the two-channel ratio analysis, as this permits comparison across different array versions. 1 µg of total RNA from each sample was labeled using the miRCURY™ LNA microRNA Power labeling Kit (Exiqon, Vedbæk, Denmark) following a two-step protocol: First, Calf Intestinal Alkaline Phosphatase (CIAP) was applied to remove

terminal 5'phosphates, and next, fluorescent labels were attached enzymatically to the 3'-end of the microRNAs. Sample specific RNA was labeled with Hy3 (green) fluorophore, while the common reference RNA pool was labeled with the Hy5 (red).

The Hy3 and Hy5 labeled RNA samples were mixed, and co-hybridized to miRCURY™ LNA Arrays (Exiqon, Vedbæk, Denmark), which contain Tm normalized capture probes targeting miRNAs from human, mouse and rat , as registered in miRBase v.15.0 at the Sanger Institute [49] . Hybridization was carried out overnight for 16 hours at 65 °C in a Tecan HS4800 hybridization station (Tecan, Männedorf, Switzerland). After washing and drying, the microarray slides were scanned under ozone free conditions (ozone level < 2.0 ppb to minimize bleaching of the fluorescent dyes) in a G2565BA Microarray Scanner System, (Agilent, Santa Clara, CA). The resulting images were quantified using Imagene v. 8.0 (BioDiscovery, El Segundo, CA), and both automatic quality control (flagging of poor spots by the software) as well as manual, visual inspection was performed to ensure the highest possible data quality.

Quantitative real-time PCR

The expression levels of 39 selected miRNAs were validated by quantitative RT-PCR applying the miRCURY LNA™ Universal RT microRNA PCR system and SYBR™ Green master mix following the manufacturer's instructions (Exiqon, Vedbæk, Denmark). The results are shown in **Supplementary Figure S1**.

Data pre-processing and normalization

All low-level analyses were carried out in the R environment, including importing and pre-processing of the data using the LIMMA package (http://bioconductor.org). Mean pixel intensities were used to calculate signal (foreground) spot intensities, and median pixel intensities were applied to estimate background intensity. After excluding flagged spots from the analysis, the "normexp" background correction method, with offset=10 was applied[50]. For intra-slide normalization, the global Lowess (LOcally Weighted Scatterplot Smoothing) regression algorithm was applied, and log2 ratios of four intra-slide replicates were averaged.

All expression data were deposited in the Rosetta Resolver (Rosetta Biosoftware, UK) data management and analysis system.

Feature selection and classification

A miRNA expression database was built for identification of miRNAs with high discriminatory power between tumor histologies. Three approaches for feature selection, filters, wrappers, and embedded methods are commonly used[51]. Here, we have applied both filtering and a wrapper: differentially expressed miRNAs were identified by running a one versus one, as well as a one versus all t-test for each histology, followed by ranking of the most significant candidate miRNAs. Additionally, the feature selection embedded in the least absolute shrinkage and selection operator (LASSO) classification algorithm was applied. We then tested and 5-fold cross-validated the LASSO algorithm and have listed its model coefficient, a measure of discriminatory potential, in **Supplementary Table S2**. The model coefficients have also been visualized in a heat-map (**Supplementary Figure S2**) The LASSO classifier was originally described by Tibshirani [52] and is based on a multinomial logistic model, which is fitted using L1 regularization[53]. The regularization parameter is chosen by evaluating the results of a cross validation along the entire regularization path. To solve the L1 regularized optimization problem. we used the glmnet algorithm[28]. The classifier was built on log2 ratio data from the 208 samples and 15 cancer classes listed in **Table 1**

Statistical analysis

All calculations and statistical tests were done in the free software environment for statistical computing and graphics R v.2.9.2 (www.r-project.org). For microarray analysis, the open source package for R, Bioconductor was used (www.bioconductor.org).

Figure 1: Expression of cancer-tissue specific miRNAs (rows) across 208 samples columns) representing the 15 histologies in the training set. The heat map shows median normalized log data for the top 5-10 miRNAs identified by LASSO's embedded feature selection algorithm) per class.

Figure 2: The plot shows the performance, assessed by 5-fold cross-validation, of the LASSO classification methods as a function of the LASSO regularization parameter: – log λ. The final regularization parameter selected for LASSO was 4.1. We settled at this value, as more complex models would entail more miRNAs without a corresponding gain in performance. The red curve shows "one-against-all" accuracy, the green curve is PPV (positive predictive value), and the blue curve is accuracy of the classifier. The "Classifier" plot (top left) represents the overall performance of the classifier. The curve shown is the percentages of correctly classified samples, while the shade indicates the standard deviation. The following 15 plots show the performance with respect to individual tissue classes. The statistics in these plots are generated in a 'one against all' fashion: i.e. the positive is the tissue class of interest, and the negative is the group of all other tissue Classes**.**

Table 1: Number of samples per tissue, true positive count (TP), mean positive predictive value (PPV), and sensitivity of the classification, assessed by 5-fold cross-validation of the classifier.

Table 2: Confusion matrix of classification results showing the number of correct classifications along the diagonal and the number of misclassifications off the diagonal (based on 5-fold cross validation of the LASSO classifier).

Table 3: Validation of the LASSO classifier on an independent test set of 48 metastatic samples. The true tissue of origin and the site of metastasis are listed in column 1 and 3, respectively. Column 2 indicates whether the classifier was correct in either its first (column 4) or second (column 6) prediction. The percentages (columns 5 and 7) are calculated by the LASSO algorithm and indicate the likelihood of a correct classification of the particular tissue.

Acknowledgments

We would like to thank Dr. Bogumil Kaczkowski and Dr. Peder Worning for the initial SVM classification of the data, and professor Wilfred D Stein for the *in silico* tissue modeling attempts. We are also grateful to Gitte Friis and Stine Jørgensen for excellent technical assistance.

Reference list

1. Pavlidis, N., Briasoulis, E., Hainsworth, J. & Greco, F.A. (2003). Diagnostic and therapeutic management of cancer of an unknown primary. *European journal of cancer (Oxford, England□: 1990)* **39**, 1990–2005

2. Pentheroudakis, G., Golfinopoulos, V. & Pavlidis, N. (2007). Switching benchmarks in cancer of unknown primary: from autopsy to microarray. *European journal of cancer (Oxford, England□: 1990)* **43**, 2026–36

3. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. & Lander, E.S. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science (New York, N.Y.)* **286**, 531–7

4. Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.H., Angelo, M., Ladd, C., Reich, M., Latulippe, E., Mesirov, J.P., Poggio, T., Gerald, W., Loda, M., Lander, E.S. & Golub, T.R. (2001). Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America* **98**, 15149–54

5. Su, A.I., Welsh, J.B., Sapinoso, L.M., Kern, S.G., Dimitrov, P., Lapp, H., Schultz, P.G., Powell, S.M., Moskaluk, C.A., Frierson, H.F. & Hampton, G.M. (2001). Molecular classification of human carcinomas by use of gene expression signatures. *Cancer research* **61**, 7388–93

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

7. Giordano, T.J., Shedden, K.A., Schwartz, D.R., Kuick, R., Taylor, J.M., Lee, N., Misek, D.E., Greenson, J.K., Kardia, S.L., Beer, D.G., Rennert, G., Cho, K.R., Gruber, S.B., Fearon, E.R. & Hanash, S. (2001). Organ-specific molecular classification of primary lung, colon, and ovarian adenocarcinomas using gene expression profiles. *The American journal of pathology* **159**, 1231–8

8. Buckhaults, P., Zhang, Z., Chen, Y.-C., Wang, T.-L., St Croix, B., Saha, S., Bardelli, A., Morin, P.J., Polyak, K., Hruban, R.H., Velculescu, V.E. & Shih, I.-M. (2003). Identifying tumor origin using a gene expression-based classification map. *Cancer research* **63**, 4144–9

9. Bloom, G., Yang, I.V., Boulware, D., Kwong, K.Y., Coppola, D., Eschrich, S., Quackenbush, J. & Yeatman, T.J. (2004). Multi-platform, multi-site, microarray-based human tumor classification. *The American journal of pathology* **164**, 9–16

10. Ma, X.-J., Patel, R., Wang, X., Salunga, R., Murage, J., Desai, R., Tuggle, J.T., Wang, W., Chu, S., Stecker, K., Raja, R., Robin, H., Moore, M., Baunoch, D., Sgroi, D. & Erlander, M. (2006). Molecular classification of human cancers using a 92-gene real-time quantitative polymerase chain reaction assay. *Archives of pathology & laboratory medicine* **130**, 465–73

11. Tothill, R.W., Kowalczyk, A., Rischin, D., Bousioutas, A., Haviv, I., van Laar, R.K., Waring, P.M., Zalcberg, J., Ward, R., Biankin, A.V., Sutherland, R.L., Henshall, S.M., Fong, K., Pollack, J.R., Bowtell, D.D.L. & Holloway, A.J. (2005). An expression-based site of origin diagnostic method designed for clinical application to cancer of unknown origin. *Cancer research* **65**, 4031–40

12. Gregory, R.I. & Shiekhattar, R. (2005). MicroRNA biogenesis and cancer. *Cancer research* **65**, 3509–12

13. Esquela-Kerscher, A. & Slack, F.J. (2006). Oncomirs - microRNAs with a role in cancer. *Nature reviews. Cancer* **6**, 259–69

14. Lu, J., Getz, G., Miska, E.A., Alvarez-Saavedra, E., Lamb, J., Peck, D., Sweet-Cordero, A., Ebert, B.L., Mak, R.H., Ferrando, A.A., Downing, J.R., Jacks, T., Horvitz, H.R. & Golub, T.R. (2005). MicroRNA expression profiles classify human cancers. *Nature* **435**, 834–8

15. Volinia, S., Calin, G.A., Liu, C.-G., Ambs, S., Cimmino, A., Petrocca, F., Visone, R., Iorio, M., Roldo, C., Ferracin, M., Prueitt, R.L., Yanaihara, N., Lanza, G., Scarpa, A., Vecchione, A., Negrini, M., Harris, C.C. &

Croce, C.M. (2006). A microRNA expression signature of human solid tumors defines cancer gene targets. *Proceedings of the National Academy of Sciences of the United States of America* **103**, 2257–61

16. Rosenfeld, N., Aharonov, R., Meiri, E., Rosenwald, S., Spector, Y., Zepeniuk, M., Benjamin, H., Shabes, N., Tabak, S., Levy, A., Lebanony, D., Goren, Y., Silberschein, E., Targan, N., Ben-Ari, A., Gilad, S., Sion-Vardy, N., Tobar, A., Feinmesser, M., Kharenko, O., Nativ, O., Nass, D., Perelman, M., Yosepovich, A., Shalmon, B., Polak-Charcon, S., Fridman, E., Avniel, A., Bentwich, I., Bentwich, Z., Cohen, D., Chajut, A. & Barshack, I. (2008). MicroRNAs accurately identify cancer tissue origin. *Nature biotechnology* **26**, 462–9

17. Rosenwald, S., Gilad, S., Benjamin, S., Lebanony, D., Dromi, N., Faerman, A., Benjamin, H., Tamir, R., Ezagouri, M., Goren, E., Barshack, I., Nass, D., Tobar, A., Feinmesser, M., Rosenfeld, N., Leizerman, I., Ashkenazi, K., Spector, Y., Chajut, A. & Aharonov, R. (2010). Validation of a microRNA-based qRT-PCR test for accurate identification of tumor tissue origin. *Modern pathology□: an official journal of the United States and Canadian Academy of Pathology, Inc* **23**, 814–23

18. Liu, A., Tetzlaff, M.T., Vanbelle, P., Elder, D., Feldman, M., Tobias, J.W., Sepulveda, A.R. & Xu, X. (2009). MicroRNA expression profiling outperforms mRNA expression profiling in formalin-fixed paraffin-embedded tissues. *International journal of clinical and experimental pathology* **2**, 519–27

19. Siebolts, U., Varnholt, H., Drebber, U., Dienes, H.-P., Wickenhauser, C. & Odenthal, M. (2009). Tissues from routine pathology archives are suitable for microRNA analyses by quantitative PCR. *Journal of clinical pathology* **62**, 84–8

20. Castoldi, M., Schmidt, S., Benes, V., Noerholm, M., Kulozik, A.E., Hentze, M.W. & Muckenthaler, M.U. (2006). A sensitive array for microRNA expression profiling (miChip) based on locked nucleic acids (LNA). *RNA (New York, N.Y.)* **12**, 913–20

21. Raphael E. Pollock, James H. Doroshow, David Khayat, Akimasa Nakao, B.O. (2005). *UICC Manual of Clinical Oncology, 8th Edition*. **33**,

22. Li, J., Smyth, P., Flavin, R., Cahill, S., Denning, K., Aherne, S., Guenther, S.M., O'Leary, J.J. & Sheils, O. (2007). Comparison of miRNA expression patterns using total RNA extracted from matched samples of formalin-fixed paraffin-embedded (FFPE) cells and snap frozen cells. *BMC biotechnology* **7**, 36

23. van Laar, R.K., Ma, X.-J., de Jong, D., Wehkamp, D., Floore, A.N., Warmoes, M.O., Simon, I., Wang, W., Erlander, M., van't Veer, L.J. & Glas, A.M. (2009). Implementation of a novel microarray-based diagnostic test for cancer of unknown primary. *International journal of cancer. Journal international du cancer* **125**, 1390–7

24. Talantov, D., Baden, J., Jatkoe, T., Hahn, K., Yu, J., Rajpurohit, Y., Jiang, Y., Choi, C., Ross, J.S., Atkins, D., Wang, Y. & Mazumder, A. (2006). A quantitative reverse transcriptase-polymerase chain reaction assay to identify metastatic carcinoma tissue of origin. *The Journal of molecular diagnostics□: JMD* **8**, 320–9

25. Shedden, K.A., Taylor, J.M.G., Giordano, T.J., Kuick, R., Misek, D.E., Rennert, G., Schwartz, D.R., Gruber, S.B., Logsdon, C., Simeone, D., Kardia, S.L.R., Greenson, J.K., Cho, K.R., Beer, D.G., Fearon, E.R. & Hanash, S. (2003). Accurate molecular classification of human cancers based on gene expression using a simple classifier with a pathological tree-based framework. *The American journal of pathology* **163**, 1985–95

26. Dennis, J.L. & Oien, K.A. (2005). Hunting the primary: novel strategies for defining the origin of tumours. *The Journal of pathology* **205**, 236–47

27. Centeno, B.A., Bloom, G., Chen, D.-T., Chen, Z., Gruidl, M., Nasir, A. & Yeatman, T.Y. (2010). Hybrid model integrating immunohistochemistry and expression profiling for the classification of carcinomas of unknown primary site. *The Journal of molecular diagnostics□: JMD* **12**, 476–86

28. Friedman, J., Hastie, T. & Tibshirani, R. (2010). Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of statistical software* **33**, 1–22

29. Dennis, J.L., Hvidsten, T.R., Wit, E.C., Komorowski, J., Bell, A.K., Downie, I., Mooney, J., Verbeke, C., Bellamy, C., Keith, W.N. & Oien, K.A. (2005). Markers of adenocarcinoma characteristic of the site of origin:

development of a diagnostic algorithm. *Clinical cancer research□: an official journal of the American Association for Cancer Research* **11**, 3766–72

30. Oien, K.A. (2009). Pathologic evaluation of unknown primary cancer. *Seminars in oncology* **36**, 8–37

31. Bloom, G.C., Eschrich, S., Zhou, J.X., Coppola, D. & Yeatman, T.J. (2007). Elucidation of a protein signature discriminating six common types of adenocarcinoma. *International journal of cancer. Journal international du cancer* **120**, 769–75

32. Dumur, C.I., Lyons-Weiler, M., Sciulli, C., Garrett, C.T., Schrijver, I., Holley, T.K., Rodriguez-Paris, J., Pollack, J.R., Zehnder, J.L., Price, M., Hagenkord, J.M., Rigl, C.T., Buturovic, L.J., Anderson, G.G. & Monzon, F.A. (2008). Interlaboratory performance of a microarray-based gene expression test to determine tissue of origin in poorly differentiated and undifferentiated cancers. *The Journal of molecular diagnostics□: JMD* **10**, 67–77

33. Bridgewater, J., van Laar, R., Floore, A. & Van'T Veer, L. (2008). Gene expression profiling may improve diagnosis in patients with carcinoma of unknown primary. *British journal of cancer* **98**, 1425–30

34. Horlings, H.M., van Laar, R.K., Kerst, J.-M., Helgason, H.H., Wesseling, J., van der Hoeven, J.J.M., Warmoes, M.O., Floore, A., Witteveen, A., Lahti-Domenici, J., Glas, A.M., Van't Veer, L.J. & de Jong, D. (2008). Gene expression profiling to identify the histogenetic origin of metastatic adenocarcinomas of unknown primary. *Journal of clinical oncology□: official journal of the American Society of Clinical Oncology* **26**, 4435–41

35. Varadhachary, G.R., Spector, Y., Abbruzzese, J.L., Rosenwald, S., Wang, H., Aharonov, R., Carlson, H.R., Cohen, D., Karanth, S., Macinskas, J., Lenzi, R., Chajut, A., Edmonston, T.B. & Raber, M.N. (2011). Prospective gene signature study using microRNA to identify the tissue of origin in patients with carcinoma of unknown primary. *Clinical cancer research□: an official journal of the American Association for Cancer Research* **17**, 4063–70

36. Varadhachary, G.R., Talantov, D., Raber, M.N., Meng, C., Hess, K.R., Jatkoe, T., Lenzi, R., Spigel, D.R., Wang, Y., Greco, F.A., Abbruzzese, J.L. & Hainsworth, J.D. (2008). Molecular profiling of carcinoma of unknown primary and correlation with clinical evaluation. *Journal of clinical oncology□: official journal of the American Society of Clinical Oncology* **26**, 4442–8

37. Geurts, P., Irrthum, A. & Wehenkel, L. (2009). Supervised learning with decision tree-based methods in computational and systems biology. *Molecular bioSystems* **5**, 1593–605

38. (Informa Healthcare: 2006). *Carcinoma of an Unknown Primary Site*. 264

39. Anderson, G.G. & Weiss, L.M. (2010). Determining tissue of origin for metastatic cancers: meta-analysis and literature review of immunohistochemistry performance. *Applied immunohistochemistry & molecular morphology□: AIMM / official publication of the Society for Applied Immunohistochemistry* **18**, 3–8

40. Pentheroudakis, G., Briasoulis, E. & Pavlidis, N. (2007). Cancer of unknown primary site: missing primary or missing biology? *The oncologist* **12**, 418–25

41. van de Wouw, A.J., Jansen, R.L.H., Speel, E.J.M. & Hillen, H.F.P. (2003). The unknown biology of the unknown primary tumour: a literature review. *Annals of oncology□: official journal of the European Society for Medical Oncology / ESMO* **14**, 191–6

42. Varadhachary, G.R. (2007). Carcinoma of unknown primary origin. *Gastrointestinal cancer research□: GCR* **1**, 229–35

43. Park, S.-Y., Kim, B.-H., Kim, J.-H., Lee, S. & Kang, G.H. (2007). Panels of immunohistochemical markers help determine primary sites of metastatic adenocarcinoma. *Archives of pathology & laboratory medicine* **131**, 1561–7

44. Ghosh, D. (2004). Mixture models for assessing differential expression in complex tissues using microarray data. *Bioinformatics (Oxford, England)* **20**, 1663–9

45. Clarke, J., Seo, P. & Clarke, B. (2010). Statistical expression deconvolution from mixed tissue samples. *Bioinformatics (Oxford, England)* **26**, 1043–9

46. Espina, V., Wulfkuhle, J.D., Calvert, V.S., VanMeter, A., Zhou, W., Coukos, G., Geho, D.H., Petricoin, E.F. & Liotta, L.A. (2006). Laser-capture microdissection. *Nature protocols* **1**, 586–603

47. Chen, L., Yan, H.-X., Yang, W., Hu, L., Yu, L.-X., Liu, Q., Li, L., Huang, D.-D., Ding, J., Shen, F., Zhou, W.-P., Wu, M.-C. & Wang, H.-Y. (2009). The role of microRNA expression pattern in human intrahepatic cholangiocarcinoma. *Journal of hepatology* **50**, 358–69

48. (panstanford: 2009). *MicroRNA Profiling in Cancer A Bioinformatics Perspective*. 23–46doi:9789814267540

49. Griffiths-Jones, S., Saini, H.K., van Dongen, S. & Enright, A.J. (2008). miRBase: tools for microRNA genomics. *Nucleic acids research* **36**, D154–8

50. Ritchie, M.E., Silver, J., Oshlack, A., Holmes, M., Diyagama, D., Holloway, A. & Smyth, G.K. (2007). A comparison of background correction methods for two-colour microarrays. *Bioinformatics (Oxford, England)* **23**, 2700–7

51. Ma, S. & Huang, J. (2008). Penalized feature selection and classification in bioinformatics. *Briefings in bioinformatics* **9**, 392–403

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

53. Bradley Efron, Trevor Hastie, Iain Johnstone, and R.T. (2004). Least Angle Regression. *Ann. Statist* **32**, 407–499

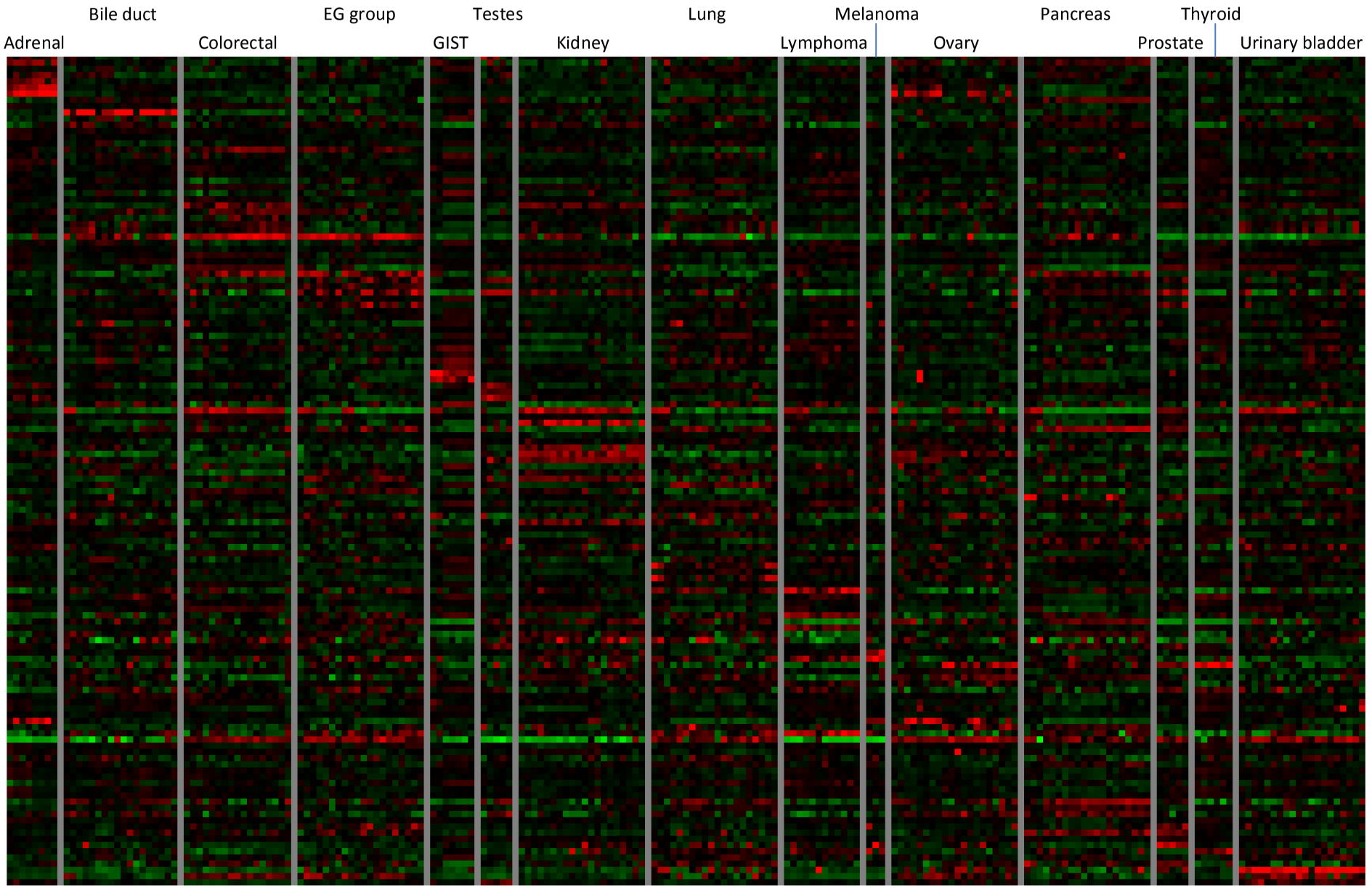


Figure 1. Expression of cancer-tissue specific miRNAs (rows) across 208 samples columns) representing the 15 histologies in the training set.
The heat map shows median normalized log data for the top 5-10 miRNAs identified by LASSO's embedded feature selection algorithm) per class.

Søkilde et al.

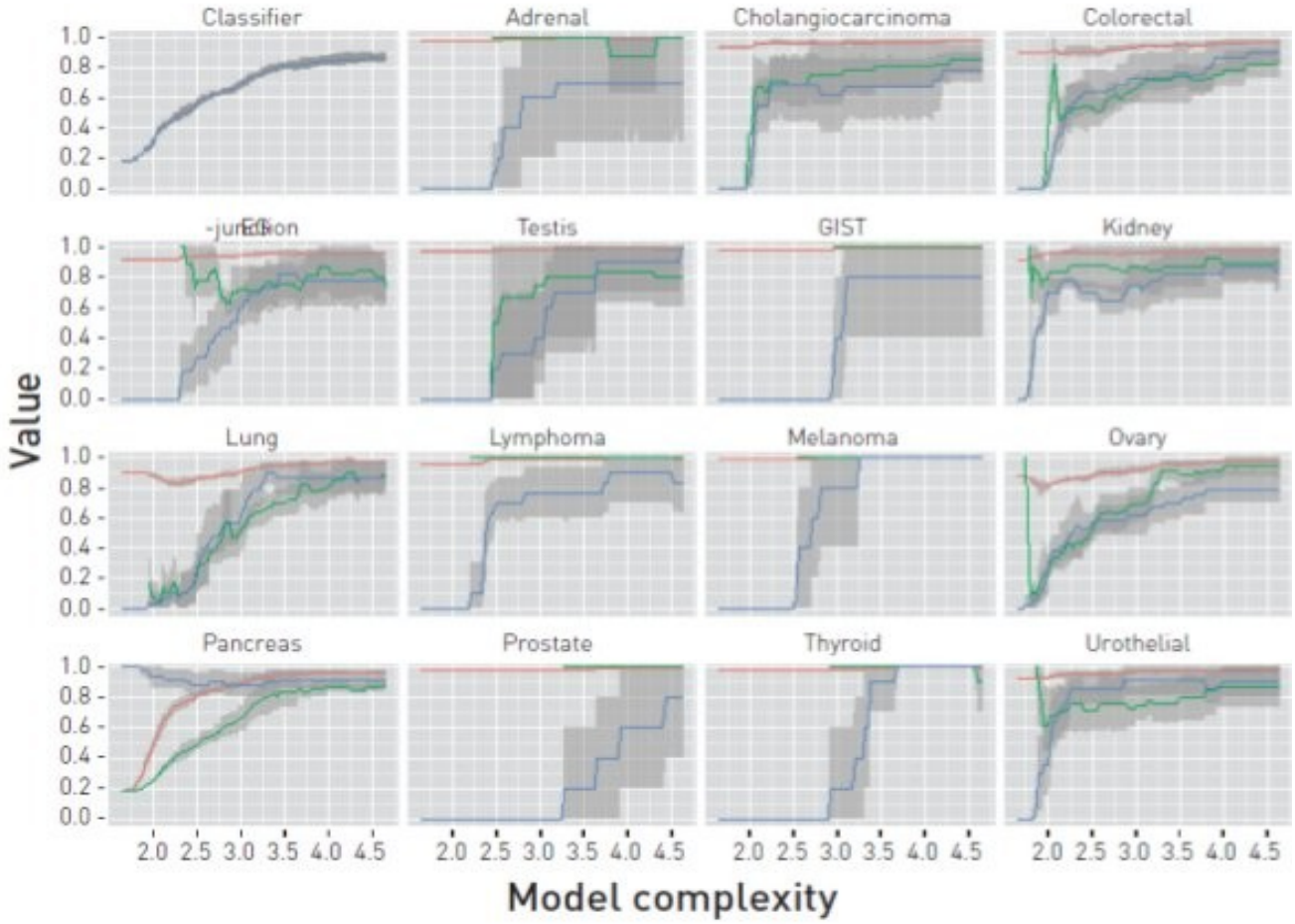

Classifier
Adrenal
Cholangiocarcinoma
Colorectal
-junction
Testis
GIST
Kidney
Lung
Lymphoma
Melanoma
Ovary
Pancreas
Prostate
Thyroid
Urothelial
Value
1.0
0.8
0.6
0.4
0.2
0.0
2.0 2.5 3.0 3.5 4.0 4.5
Model complexity

Figure 2.

Tissue	Histology	Samples (n)	TP	Mean PPV	Mean Sensitivity
Adrenal	Cortical carcinoma (ACC)	8	6	100%	70%
Bile duct	Cholangiocarcinoma	18	14	100%	78%
Colorectal	Adenocarcinoma, Mucinous adenocarcimona	17	13	77%	78%
[1] EG-junction	Adenocarcinoma, signet cell, mucinous adenocarcinoma, (Squamous excluded)	20	17	83%	85%
Germ cell tumor	Non-seminoma, seminoma, embryonal carcinoma, yolk sac carcinoma	7	7	83%	100%
GIST	Gastrointestinal stromal tumor	5	4	100%	80%
Kidney	Papillary cell carcinoma, clear cell carcinoma	20	18	87%	90%
Lung	Adenocarcinoma (Squamous excluded)	20	18	86%	90%
Lymphoma	B cell, large cell, marginal zone Hodgkin's	13	12	95%	93%
Melanoma	Malignant melanoma	9	9	100%	100%
Ovary	Serous, mucinous, endometrioid adenocarcinoma, clear cell	20	13	90%	65%
Pancreas	Ductal adenocarcinoma, mucinous non-cystic	20	16	80%	80%
Prostate	Adenocarcinoma	5	4	100%	80%
Thyroid	Papillary, Hurthle cell, follicular carcinoma	6	6	100%	100%
Urinary bladder	Transitional cell carcinoma, papillary and non-papillary	20	19	83%	95%
Total		208	176		

Table 1: Number of samples per tissue, true positive count (TP), mean positive predictive value (PPV), and sensitivity of the classification, assessed by 5-fold cross-validation of the classifier.

[1] The "EG-junction" class combines samples from esophagus and gastric cancers.

		True class														
		Adrenal gland	Cholangiocarcinoma	Colorectal	EG - junction	Germ cell tumor	GIST	Kidney	Lung	Lymphoma	Melanoma	Ovary	Pancreas	Prostate	Thyroid	Urothelial
Predicted class	Adrenal gland	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Cholangiocarcinoma	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0
	Colorectal	0	0	13	3	0	0	0	0	0	0	1	1	0	0	0
	EG – junction	0	1	2	17	0	0	0	0	0	0	1	2	0	0	0
	Germ cell tumor	0	0	0	0	7	0	0	0	0	0	1	0	0	0	1
	GIST	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0
	Kidney	0	1	0	0	0	1	18	0	0	0	1	1	0	0	0
	Lung	0	2	0	0	0	0	0	18	0	0	1	0	0	0	0
	Lymphoma	0	0	0	0	0	0	0	1	12	0	0	0	0	0	0
	Melanoma	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0
	Ovary	1	0	0	0	0	0	1	0	0	0	13	0	0	0	0
	Pancreas	1	0	2	0	0	0	1	0	0	0	1	16	0	0	0
	Prostate	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0
	Thyroid	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0
	Urothelial	0	0	0	0	0	0	0	1	1	0	1	0	1	0	19

Table 2: Confusion matrix of classification results showing the number of correct classifications along the diagonal and the number of mis-classifications off the diagonal (based on 5-fold cross validation of the LASSO classifier).

True Class	Correct	Metastasis site	1. Prediction	Percent	2. prediction	Percent
Colorectal	2nd	Pelvis	*EG – junction	31%	Colorectal	22%
Colorectal	1st	Adrenal gland	Colorectal	52%	Ovary	16%
Colorectal	1st	Liver	Colorectal	74%	Ovary	11%
Colorectal	1st	Liver	Colorectal	51%	EG – junction	20%
Colorectal	1st	Liver	Colorectal	81%	Ovary	7%
Colorectal	1st	Liver	Colorectal	70%	Ovary	9%
Colorectal	No	Liver	Pancreas	30%	Kidney	17%
Colorectal	1st	Lung	Colorectal	54%	EG – junction	14%
Colorectal	1st	Liver	Colorectal	61%	EG – junction	11%
Colorectal	2nd	Omentum	Pancreas	35%	Colorectal	22%
Colorectal	2nd	Liver	EG – junction	30%	Colorectal	19%
Colorectal	2nd	Omentum	EG – junction	40%	Colorectal	37%
Colorectal	1st	Lung	Colorectal	53%	EG – junction	17%
Colorectal	1st	Pending	Colorectal	56%	EG – junction	23%
EG - junction	1st	Lymph node	EG – junction	17%	Pancreas	15%
EG - junction	1st	Lymph node	EG – junction	54%	Lymphoma	19%
EG - junction	1st	Lymph node	EG – junction	46%	Colorectal	14%
Pancreas	1st	Lymph node	Pancreas	81%	Lung	3%
Pancreas	2nd	Omentum	EG – junction	19%	Pancreas	17%
Pancreas	2nd	Abdominal wall	Lung	17%	Pancreas	16%
Pancreas	No	Liver	Colorectal	51%	EG – junction	22%
Pancreas	No	Omentum	EG – junction	40%	Colorectal	31%
Ovary	1st	Bowel	Ovary	37%	Urothelial carcinoma	23%
Ovary	1st	Colon	Ovary	31%	Lung	22%
Ovary	2nd	Colon	Pancreas	60%	Ovary	13%
Ovary	1st	Colon	Ovary	94%	Thyroid	4%
Ovary	No	Colon	EG – junction	46%	Pancreas	9%
Ovary	1st	Gastric wall	Ovary	38%	Thyroid	38%
Ovary	1st	Omentum	Ovary	33%	Lung	18%
Ovary	1st	Omentum	Ovary	37%	Pancreas	26%
Ovary	1st	Omentum	Ovary	45%	Thyroid	12%
Ovary	1st	Omentum	Ovary	70%	Kidney	19%
Ovary	No	Omentum	Urothelial	49%	Lung	32%
Ovary	1st	Omentum	Ovary	83%	Pancreas	6%
Ovary	2nd	Omentum	Cholangiocarcinoma	19%	Ovary	16%
Ovary	1st	Omentum	Ovary	55%	Thyroid	17%
Ovary	1st	Omentum	Ovary	47%	Thyroid	18%
Ovary	2nd	Pelvis	Lung	39%	Ovary	16%
Ovary	1st	Pending	Ovary	31%	Lung	13%
Ovary	1st	Pending	Ovary	54%	Thyroid	16%
Kidney	1st	Lung	Kidney	51%	Cholangiocarcinoma	14%
Kidney	1st	Lymph node	Kidney	61%	Cholangiocarcinoma	10%
Kidney	1st	Adrenal gland	Kidney	76%	Melanoma	4%
Kidney	1st	Pancreas	Kidney	96%	EG – junction	1%
Kidney	1st	Pancreas	Kidney	42%	Ovary	29%
Lung	1st	Lymph node	Lung	44%	Kidney	12%
Lung	No	Lymph node	Urothelial	48%	Cholangiocarcinoma	30%
Urothelial	1st	Colon	Urothelial	75%	Pancreas	13%

Table 3: Validation of the LASSO classifier on an independent test set of 48 metastatic samples. The true tissue of origin and the site of metastasis are listed in column 1 and 3, respectively. Column 2 indicates whether the classifier was correct in either its first (column 4) or second (column 6) prediction. The percentages (columns 5 and 7) are calculated by the LASSO algorithm and indicate the likelihood of a correct classification of the particular tissue.

Nomenclature

ι_I	The canonical inclusion into the I'th parameter group, page 42
β	A parameter, i.e an element of B, page 10
$\beta^{(I)}$	Group I of β - a vector, page 23
$\mathfrak{D}$	A random data set, page 11
Δ^K	The K'th probability simplex, page 6
Err	The expected generalization error, page 22
$\mathrm{err}(\mathfrak{p})$	The generalization error of $\mathfrak{p}$, page 8
η	The linear predictors, page 14
Grp	Number of nonzero groups, page 23
$\hat{R}_{\mathfrak{D}}$	The empirical risk, page 11
$\hat{\beta}$	Parameter estimator, page 20
$\hat{\beta}_N$	Parameter estimator for n samples, page 20
$\mathbb{N}$	The positive natural numbers, page 6
$\mathbb{R}_+$	The positive real numbers, page 15
$\mathcal{A}$	Supervised learning method, page 20
$\mathcal{A}(\lambda)$	Parametrized supervised learning method, page 24
$\mathcal{A}_N$	Supervised learning method for n samples, page 20
$\mathcal{P}$	Subset of the set of all conditional distributions on Y given X, page 134
$\mathcal{S}_K$	The finite set containing K elements, page 6
$\mathfrak{p}$	A classifier, page 6
$\mathfrak{p}^{\mathrm{Bayes}}$	The Bayes classifier, page 7
Par	Number of nonzero parameters, page 23
Φ	Sublinear penalty, page 36

σ_C Support function of C, page 136

A Positive definite matrix, page 43

B The set of parameters, page 10

C A convex set, page 136

C_T Model characteristic, page 22

D A realization of a random data set, page 11

D Data set, realization of a random data set $\mathfrak{D}$, page 20

F Joint distribution of (X, Y), page 6

$f(x, y)$ The joint distribution of (X, Y), page 6

H Hessian matrix, page 43

h A linear model - a function $\mathbb{R}^K \to \Delta^K$, page 14

H_{II} Diagonal block of the Hessian matrix corresponding to the I'th group, page 43

K The number of classes, page 6

L A loss function, page 7

N The number of samples, page 11

$N_C(x)$ Normal cone to C at x - a cone, page 137

p The dimension of the covariate vector, page 6

P_C The projection onto compact convex set C - a function, page 137

P_I Projection operator, part of a decomposition, page 38

R risk, page 11

X The covariate vector - a random vector, page 6

Y The response vector - a discrete random vector, page 6

Index

Bibliography

[1] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4(0):40–79, 2010.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Berichte über verteilte messysteme. Cambridge University Press, 2004. ISBN 9780521833783.

[4] M. H. Cheok, W. Yang, C. H. Pui, J. R. Downing, C. Cheng, C. W. Naeve, M. V. Relling, and W. E. Evans. Treatment-specific changes in gene expression discriminate in vivo drug response in human leukemia cells. *Nat Genet*, 34(1):85–90, May 2003. ISSN 1061-4036. doi: 10.1038/ng1151. URL `http://dx.doi.org/10.1038/ng1151`.

[5] H. Cramér. *Mathematical Methods of Statistics*. Princeton Landmarks in Mathematics and Physics Series. PRINCETON University Press, 1945. ISBN 9780691005478.

[6] A. Frank and A. Asuncion. UCI machine learning repository, 2010. URL `http://archive.ics.uci.edu/ml`.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer series in statistics. Springer-Verlag New York, 2009. ISBN 9780387848587.

[8] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. pages 1137–1143. Morgan Kaufmann, 1995.

[9] S. Liu, Z. Liu, J. Sun, and L. Liu. Application of synergetic neural network in online writeprint identification. *International Journal of Digital Content Technology and its Applications*, 5(3):126–135, 2011.

[10] J. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley Series in Probability and Statistics. Wiley, 1999. ISBN 9780471986331.

[11] L. Meier, S. V. D. Geer, P. Bühlmann, and E. T. H. Zürich. The group lasso for logistic regression. *Journal of the Royal Statistical Society, Series B*, 2008.

[12] K. Perell, M. Vincent, B. Vainer, B. L. Petersen, B. Federspiel, A. K. Møller, M. Madsen, L. F. Hansen, N. R. Hansen, F. C. Nielsen, and G. Daugaard. A microrna-based primary tumor site classification of liver core-biopsies. *Clinical Cancer Research*.

[13] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. Mclaughlin, J. Y. H. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415 (6870):436–442, Jan. 2002.

[14] R. Søkilde, M. Vincent, A. K. Møller, A. Hansen, P. E. Høiby, T. Blondal, B. S. Nielsen, G. Daugaard, S. Møller, and T. Litman. Efficient identification of metastases by their microrna profile.

[15] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[16] P. Tseng and C. O. L. Mangasarian. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim Theory Appl*, pages 475–494, 2001.

[17] P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Math. Program.*, 117(1-2):387–423, 2009.

[18] J. Urruty and C. Lemaréchal. *Fundamentals of convex analysis*. Grundlehren Text Editions Series. Springer Verlag, 2001. ISBN 9783540422051.

[19] V. Vapnik. *The Nature of Statistical Learning Theory*. Information Science and Statistics. Springer, 2000. ISBN 9780387987804.

[20] M. Vincent. *msgl: Multinomial sparse group lasso*, 2013. URL `http://cran.r-project.org/web/packages/msgl/index.html`. R package version 0.1.3.

[21] M. Vincent and N. R. Hansen. Sparse group lasso and high dimensional multinomial classification. *Computational Statistics & Data Analysis*, May 2012. URL `http://arxiv.org/abs/1205.1245`.

[22] M. Vincent, K. Perell, F. C. Nielsen, G. Daugaard, and N. R. Hansen. Modeling tissue contamination to improve molecular identification. *Bioinformatics*.

[23] L. Wasserman. *All of Statistics: A Concise Course in Statistical Inference*. Springer Publishing Company, Incorporated, 2010. ISBN 1441923225, 9781441923226.

[24] J. Zhu and T. Hastie. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(14):427–443, 2004.