# Cox regression with high-dimensional covariates

Ørnulf Borgan
Department of Mathematics
University of Oslo

NORBIS course
University of Oslo
4-8 December 2017



## Outline

- Dutch breast cancer data
- Data and model
- Overview of methods for high-dimensional data
  - Variable selection
  - Dimension reduction
  - Penalized regression
- Penalized Cox regression
- Cross-validation
- Cox-lasso for the Dutch breast cancer data



## Dutch breast cancer data

In this lecture we will for illustration use a data set on survival for $n$ = 295 breast cancer patients diagnosed in the Netherlands between 1984 and 1995 (and who fulfilled certain criteria; cf. van Houwelingen et al. *Statist in Medicine*, 2006).

For each of the patients we have $p$ = 4919 cDNA microarray gene expression measurements (selected by some procedure from the original 24885 genes measured)

79 of the women were recorded to die during the period of follow-up, and 216 were censored (the median time of follow-up was 7.2 years)

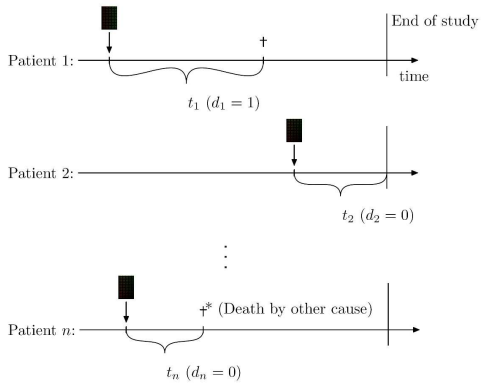
We will study how the gene expression data may be used to predict the survival of these women



## Data and problem

For each of $n$ patients we have:

- Measurements of $p$ genomic variables ($p >> n$)
  E.g. microarray gene expressions, copy numbers or SNPs
- A censored survival time



| Patient # | Survival time | cens. indicator | Genetic measurements | | | |
|---|---|---|---|---|---|---|
| 1 | 10.1 | 1 | 0.01 | 2.10 | ⋯ | 0.00 |
| 2 | 8.1 | 0 | -0.01 | 0.22 | ⋯ | 0.01 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $n$ | 5.0 | 0 | -0.7 | 0.63 | ⋯ | 0.04 |

The aim is to identify the genomic variables that may be influential for survival and to build a model that may predict survival of a patients from her genomic variables

(One will often have clinical covariates in addition to the genomic variables, but that will not be considered here.)



## Cox proportional hazards model

Hazard rate for patient $i$ is given as

$$\alpha(t \mid \mathbf{x}_i) = \alpha_0(t)\exp\{\beta_1 x_{i1} + \ldots + \beta_p x_{ip}\} = \alpha_0(t)\exp\{\boldsymbol{\beta}^T \mathbf{x}_i\}$$

The logarithm of Cox's partial likelihood is given by

$$\ell(\boldsymbol{\beta}) = \sum_{T_j}\left\{\boldsymbol{\beta}^T \mathbf{x}_{i_j} - \log\left(\sum_{l \in \mathcal{R}_j} \exp\left(\boldsymbol{\beta}^T \mathbf{x}_l\right)\right)\right\}$$

Here $i_j$ is the index of the patient who dies at $T_j$, and $\mathcal{R}_j$ is the risk set at $T_j$

When $p < n$, $\hat{\boldsymbol{\beta}}$ is obtained by unconstrained maximization of the log partial likelihood (as discussed earlier)

When $p >> n$, some regularized estimation technique is required



## Overview of methods for high-dimensional data

### Variable selection

- Univariate selection
- Forward stepwise selection
- Boosting

### Dimension reduction

- Principal component regression
- Partial least squares regression

### Penalized regression

- Ridge regression
- Lasso
- Elastic net

All the methods depend on a «tuning parameter» $\lambda$ (or two tuning parameters)



## Variable selection

### Univariate selection

- Each gene is tested individually for association with survival by including it into a univariate Cox regression
- The subset of $\lambda$ genes with the smallest *P*-values are included in a Cox regression model (often determined using «false discovery rate»)

### Forward stepwise selection

- First the most significant gene is selected
- Then all models with this gene and one more gene are considered. The new gene from the most significant model is selected, and included in the model
- This procedure is continued until $\lambda$ genes are included

### Boosting

- Boosting is a version of forward stepwise regression, where one only does a partial update in each step

## Dimension reduction methods

### Principal component regression (PCR)

- The high dimensional genomic measurements are decomposed using principal component analysis
- The first $\lambda$ principal components are included as covariates in a Cox regression model
- The survival times are not used to guide the choice of principal components, so no special theory is needed for Cox regression

### Partial least squares (PLS)

- PLS regression for linear regression models performs regression of the outcome on a small number of components which are linear combinations of the original covariates (but unlike PCR these components depend on the outcome)
- The Cox model is not linear, but modifications of PLS have been developed for Cox regression



## Penalized Cox regression

- Shrink the estimated regression coefficients towards zero by imposing a penalty large on coefficient values (all covariates should be on the «same scale»)
- In the Cox regression setting the coefficients are estimated by maximizing a penalized log partial likelihood of the form

$$\ell(\boldsymbol{\beta}) - \lambda \sum_{j=1}^{p} \text{pen}(\beta_j)$$

- For ridge regression we have $\text{pen}(\beta_j) = \beta_j^2$
- For lasso we have $\text{pen}(\beta_j) = |\beta_j|$
- For elastic net we have $\text{pen}(\beta_j) = \alpha|\beta_j| + (1-\alpha)\beta_j^2$

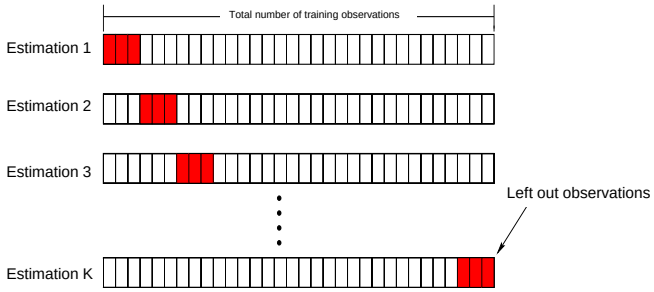
Lasso and elastic net perform variable selection, since many of the estimated coefficients will be set to zero

## Cross-validation

We have to determine the value of the «tuning parameter» $\lambda$

This may be done by K-fold cross-validation:

1) Split the data into K parts. Leave part *j* out, and use the other K-1 parts of the data to fit the model (with a given value of $\lambda$)
   Denote the estimate thus obtained by $\hat{\boldsymbol{\beta}}_{(-j)}(\lambda)$
2) Repeat for all folds *j* = 1, …, K.






3) Define the K-fold cross-validated log partial likelihood by

$$CV(\lambda) = \sum_{j=1}^{K} \ell_j(\hat{\boldsymbol{\beta}}_{(-j)}(\lambda))$$

where

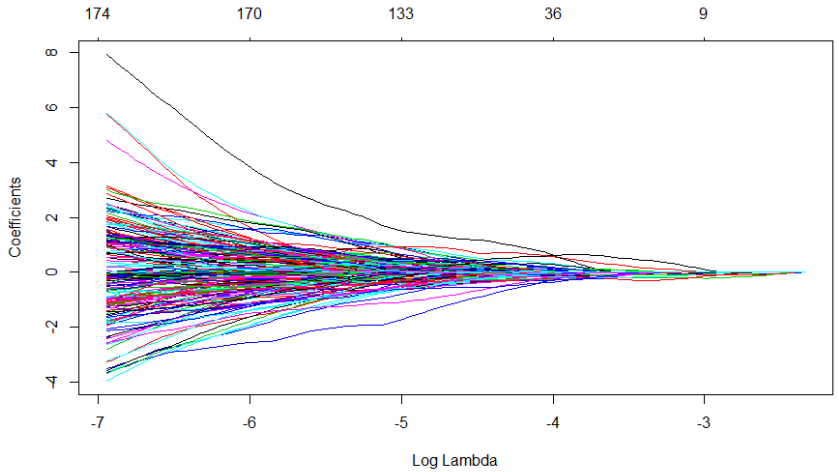
$$\ell_j(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) - \ell_{(-j)}(\boldsymbol{\beta})$$

is the contribution of the *j*-th fold to the partial likelihood

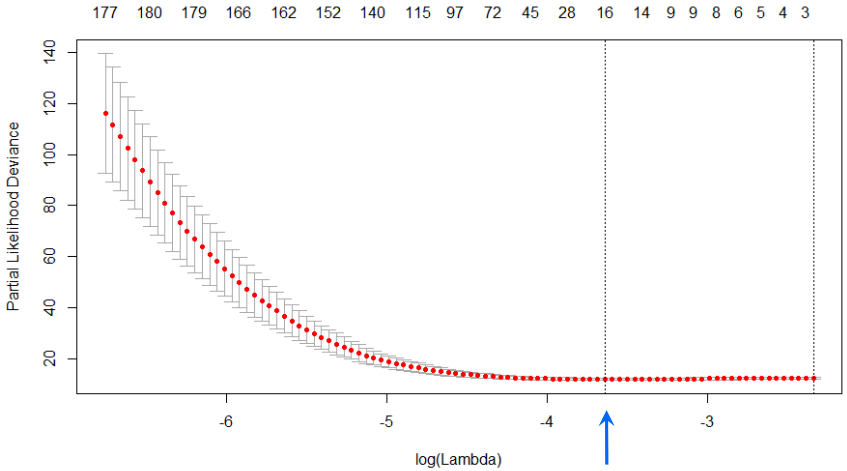
4) The optimal tuning parameter $\lambda_{opt}$ is obtained by maximizing $CV(\lambda)$

## Lasso for Dutch breast cancer data

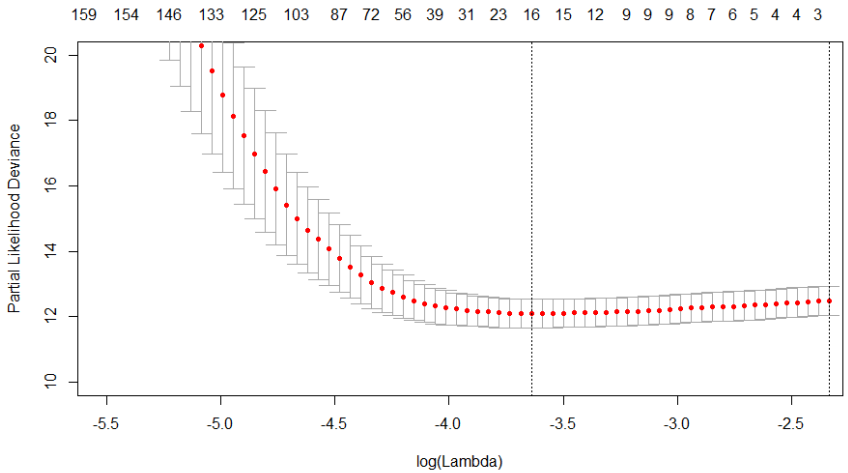
Plot of the estimates obtained for various values of $\lambda$ (for the purpose of illustration)




The cross-validation curve $CV(\lambda)$ based on 10-fold cross-validation

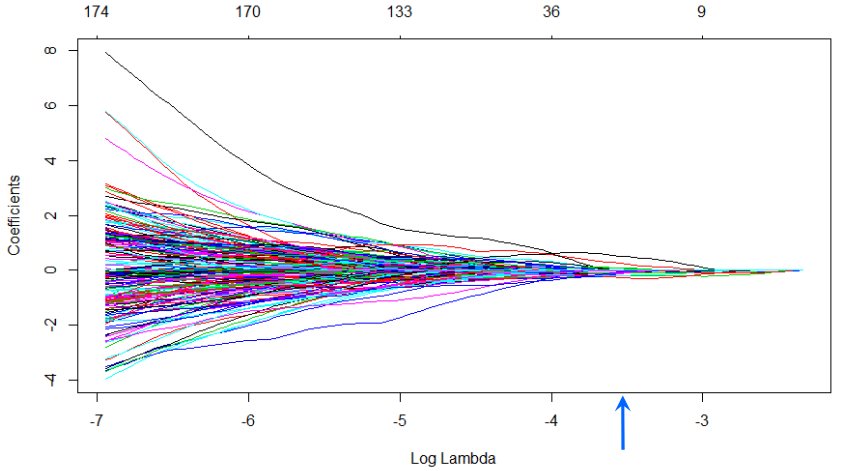



$\log \lambda_{\text{opt}} = -3.64$




(glmnet scales the log-likelihood by $n$, so for the formulas given earlier we have $\lambda_{\text{opt}} = 295 \cdot e^{-3.64} = 7.74$)

With the optimal value of $\lambda$ we obtain 16 estimates that are different from zero




$\log \lambda_{\text{opt}} = -3.64$

## Using R

The results on the previous slides are obtained by the commands below.

```
# # Read breast cancer data and extract variables
# (It takes quite some time to read the data)
breastcancer=read.table("http://folk.uio.no/borgan/IMB9335/breast.cancer.txt")

# For our illustration we want to fix the K=10 folds
# We do this by first permuting randomly the rows of the data matrix
# and then assigning the folds 1,2,....,10 cyclically to the permuted rows
set.seed(113457)
ind=sample(1:295,295)
breastcancer=breastcancer[ind,]
fold=rep(1:10,length.out=295)

# Extract the (censored) survival times, statuses and gene expressions:
time=breastcancer[,1]
status=breastcancer[,2]
geneexpr=as.matrix(breastcancer[,-(1:2)])
```



```
# We use the survival and glmnet-packages
library(survival); library(glmnet)

# Fit Cox regression using lasso with 10-fold cross-validation
cox.lasso=cv.glmnet(geneexpr,Surv(time,status),family="cox",
                    foldid=fold, standardize=FALSE)

# First plot estimates as a function of lambda:
plot(cox.lasso$glmnet.fit,xvar="lambda")

# Plot cross-validation curve
plot(cox.lasso)

# Value of lambda that gives the minimum of the cross-validation curve
cox.lasso$lambda.min

# Estimated coefficients:
coefficients=coef(cox.lasso, s=cox.lasso$lambda.min)
active.index=which(coefficients != 0)
active.coefficients=coefficients[active.index]
active.coefficients
covarno=predict(cox.lasso, s=cox.lasso$lambda.min,type="nonzero")
cbind(covarno,active.coefficients)
```

The lasso give 16 estimated coefficients that are different from zero (since we do not know their names/functions, we do not consider the estimates here)

Note that a different list of genes may give about the same prediction performance, and that another split into 10 folds will not give the same list of genes

If one wants to find the genes that are of real importance, it may be useful to focus the genes that are selected in «most splits» of the data into 10 folds