Prediction of ER+ breast cancer using gradient descent logistic regression

By Georgina Gonzalez. May 30th, 2019

Introduction

Copy Number Aberrations, gains and losses of genomic regions, are a hallmark of cancer. Copy number data is high-dimensional and is characterized by heavy correlated features. Often, like in this case, the number of samples is small compared to the number of features. In this work I first reduce the dimensionality using Topological Analysis of array CGH (TAaCGH) [1] detecting regions of the genome with significant aberrations in copy number for patients with over-expression in estrogen receptor (ER+). Next it is determined if each of the patients is aberrant for those particular regions creating, as a result, a set of binary variables that will be used as features in a logistic regression model to predict ER+ breast cancer [2].

This is a companion text to the scripts that produce the gradient descent logistic regression model for ER+.

References

[1] Daniel DeWoskin, Joan Climent, I Cruz-White, Mariel Vazquez, Catherine Park, and Javier Arsuaga. Applications of computational homology to the analysis of treatment response in breast cancer patients. Topology and its Applications, 157(1):157–164, 2010.

[2] Gonzalez G, Ushakova A, Sazdanovic R, Arsuaga J. Prediction in cancer genomics using topological signatures and machine learning. The Abel Symposium "Topological Data Analysis" 2018At: Geiranger, NorwayVolume: (in Press).

Climent data set: Joan Climent, Peter Dimitrow, Jane Fridlyand, Jose Palacios, Reiner Siebert, Donna G Albertson, Joe W Gray, Daniel Pinkel, Ana Lluch, and Jose A Martinez-Climent. Deletion of chromosome 11q predicts response to anthracycline-based chemotherapy in early breast cancer. Cancer research, 67(2):818–826, 2007.

Horlings data set: Hugo M Horlings, Carmen Lai, Dimitry SA Nuyten, Hans Halfwerk, Petra Kristel, Erik van Beers, Simon A Joosse, Christiaan Klijn, Petra M Nederlof, Marcel JT Reinders, et al. In- tegration of dna copy number alterations and prognostic gene expression signatures in breast cancer patients. Clinical Cancer Research, 16(2):651–663, 2010.

The scripts

- sigmoid.R - sigmoid function
- cost_reg_logit.R - cost function for logistic regression
- predict.R - prediction function
- score.R - score for logistic regression: F1 or Accuracy
- grad_reg_logit.R - gradient descent for one lambda
- grad_reg_logit_optim_iterLambda.R - gradient descent for a vector of lambdas
- curveLambdaVSscore.R - plot score (F1 or Accuracy) for every lambda
- plot_thetaVSlambda.R - plot coefficients vs lambda
- learningCurve.R - plot learning curve for the final model

Training data set: Horlings

The training data set is Horlings and consists of 66 samples. After applying TAaCGH to the ER+ phenotype, only 10 regions resulted significant and a binary variable was created for each of them where 1 means that the aberration is present in the sample. Features do not have any missing values but the response variable ER+ does. These are the frequencies after removing missing values

Response variable: Over-expression of estrogen receptor (ER+)

```
table(trainSet$ERpos)
```

```
##
## ER- ER+
##  28  38
```

Frequencies for significant regions after TAaCGH (features):

```
# Frequency table for features
sapply(trainSet[,4:13],table)
```

```
##              ERpos_5pseg3_sig ERpos_CM_16p_sig ERpos_CM_16q_sig
## Non-Aberrant               31               37               15
## Aberrant                   35               29               51
##              ERneg_CM_2p_sig ERneg_CM_4p_sig ERneg_CM_4q_sig
## Non-Aberrant              29              28              17
## Aberrant                  37              38              49
##              ERneg_CM_5q_sig ERneg_CM_6p_sig ERneg_CM_10q_sig
## Non-Aberrant              19              32               11
## Aberrant                  47              34               55
##              ERneg_CM_14q_sig
## Non-Aberrant               16
## Aberrant                   50
```

Validation data set: Climent

It is common with genomic arrays that the platform or the laboratory might have an effect on the data so it is best not to mix them. I chose to keep the full Climent data set as validation set which consists of 161 samples. Features do not have any missing values but the response variable ER+ does. These are the frequencies after removing missing values

Response variable: Over-expression of estrogen receptor (ER+)

```
table(valSet$ERpos)
```

```
##
## ER- ER+
##  60 101
```

Frequencies for significant regions after TAaCGH (features):

```
# Frequency table for features
sapply(valSet[,4:13],table)
```

```
##                ERpos_5pseg3_sig ERpos_CM_16p_sig ERpos_CM_16q_sig
## Non-Aberrant                 65               37               43
## Aberrant                     96              124              118
##                ERneg_CM_2p_sig ERneg_CM_4p_sig ERneg_CM_4q_sig
## Non-Aberrant               102              73              58
## Aberrant                    59              88             103
##                ERneg_CM_5q_sig ERneg_CM_6p_sig ERneg_CM_10q_sig
## Non-Aberrant                63             110               70
## Aberrant                    98              51               91
##                ERneg_CM_14q_sig
## Non-Aberrant                 59
## Aberrant                    102
```

Running gradient descent with regularized logistic regression

Logistic regression hypothesis

$h_\theta(x) = g(\theta^T x)$

where g is the sigmoid function: $g(z) = \frac{1}{1+e^{-z}}$.

The cost function

$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[-y^{(i)}log(h_\theta(x^{(i)})) - (1-y^{(i)})log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta^2$
.

Gradient

$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$,

where m is the number of samples, n the number of features and y corresponds to the response variable.

Cost and gradient for initial_theta equal to zero

```
# Compute and display initial cost
J <- cost_reg_logit(initial_theta, X, y, lambda);
J %>% round(3) %>% paste('Cost at initial theta (zeros):') %>% print()
```

```
## [1] "0.693 Cost at initial theta (zeros):"
```

Gradient descent with no regularization (lambda=0)

```
# Gradient descent with optim
optimized_noReg <- optim(par=initial_theta,X=X,y=y,lambda=lambda, fn=cost_reg_logit,gr=grad_reg_logit)
print("The optimized theta values with no regularization (lambda=0) are: ")
```

```
## [1] "The optimized theta values with no regularization (lambda=0) are: "
```

```
theta <- optimized_noReg$par
names(theta) <- c("Intercept", colnames(trainSet[,inputVars]))
print(round(theta,3))
```

```
##        Intercept ERpos_5pseg3_sig ERpos_CM_16p_sig ERpos_CM_16q_sig
##            3.126            0.091            0.448            2.419
##  ERneg_CM_2p_sig  ERneg_CM_4p_sig  ERneg_CM_4q_sig  ERneg_CM_5q_sig
##           -1.219           -2.938           -1.708            0.713
##  ERneg_CM_6p_sig ERneg_CM_10q_sig ERneg_CM_14q_sig
##           -1.476           -0.433           -0.173
```

```
print('The cost at the final theta values with no regularization (lambda=0) is: ')
```

```
## [1] "The cost at the final theta values with no regularization (lambda=0) is: "
```

```
print(round(optimized_noReg$value,3))
```

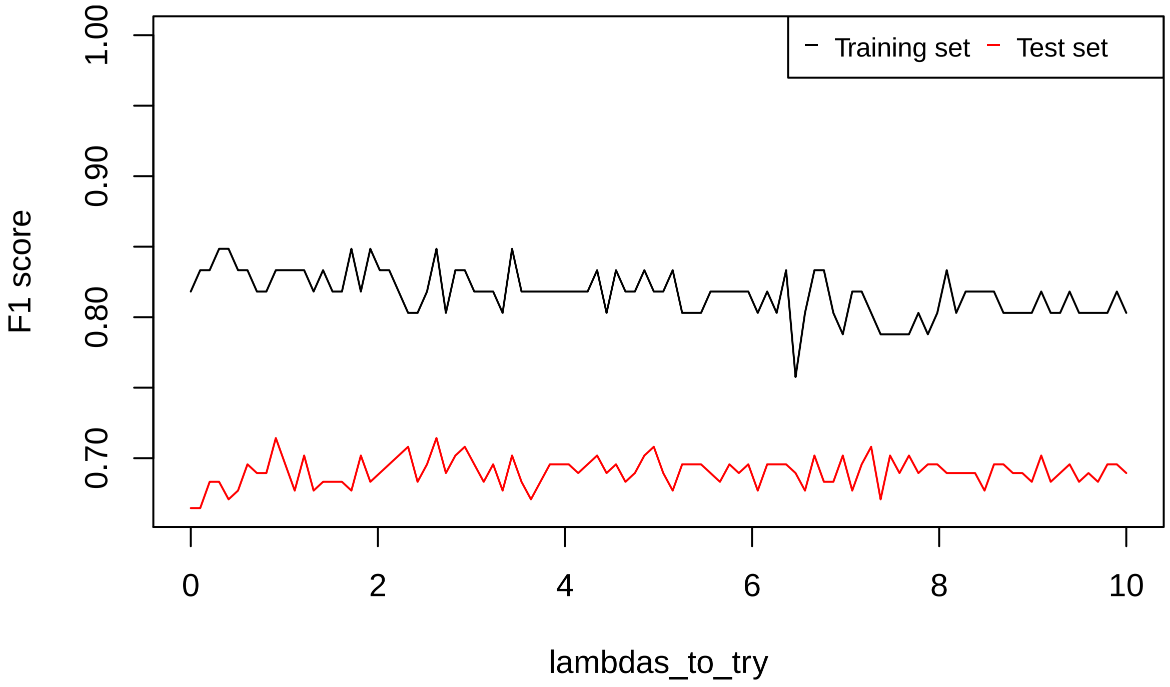
```
## [1] 0.346
```

Gradient descent with regularization

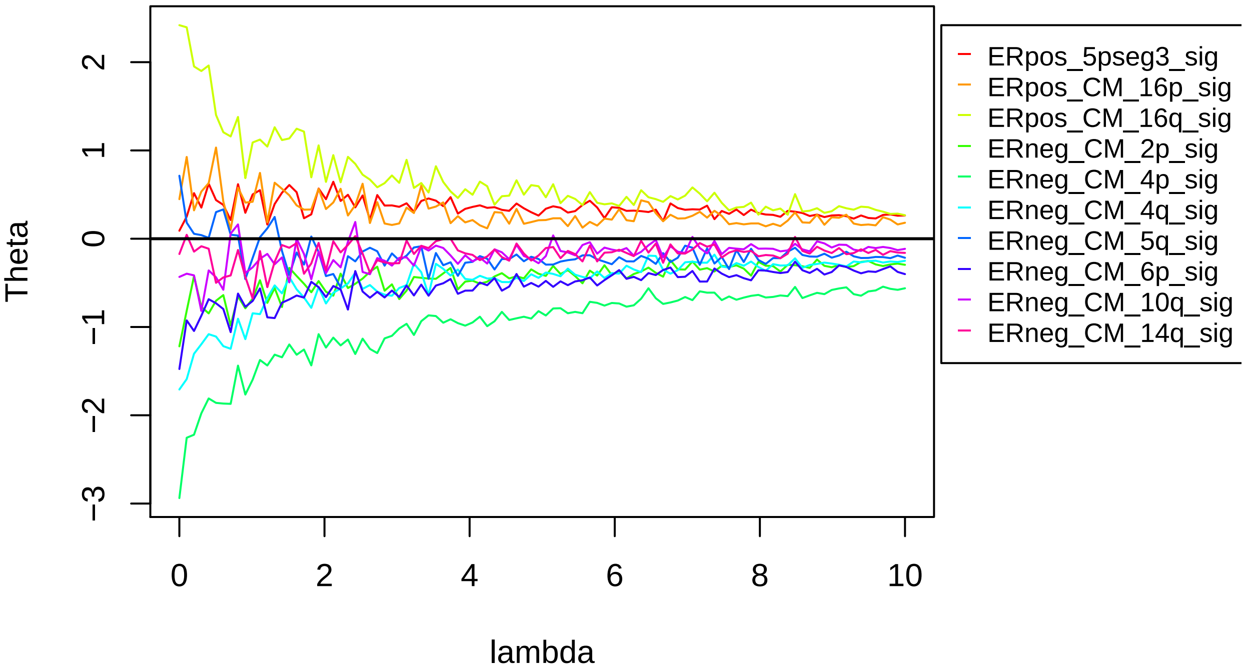
Choosing the right penalty in the regularization (lambda)

```
# Choosing lambda
lambdas_to_try <- seq(0, 10, length.out = 100)
optimObj <- grad_reg_logit_optim_iterLambda(initial_theta,  X, y, lambdas_to_try, metric="F1")
thetaMat <- optimObj$thetaMat
colnames(thetaMat) <- c("Intercept", colnames(trainSet[,inputVars]))

scoreVal <- curveLambdaVSscore(thetaMat, X, y, Xval, yval, metric)
```



```
 plot_thetaVSlambda(thetaMat, lambdas_to_try)
```



```
 best.lambda.idx <- which.max(scoreVal)
 best.lambda <- lambdas_to_try[best.lambda.idx]
 final.scoreVal <- max(scoreVal)*100
paste('The best', metric, 'score is:', round(final.scoreVal,1),'% at lambda=', round(best.lambda,3)) %>
```

```
## [1] "The best F1 score is: 71.4 % at lambda= 0.909"
```

```
paste('The optimized theta values for lambda=', round(best.lambda,3), 'are:')
```

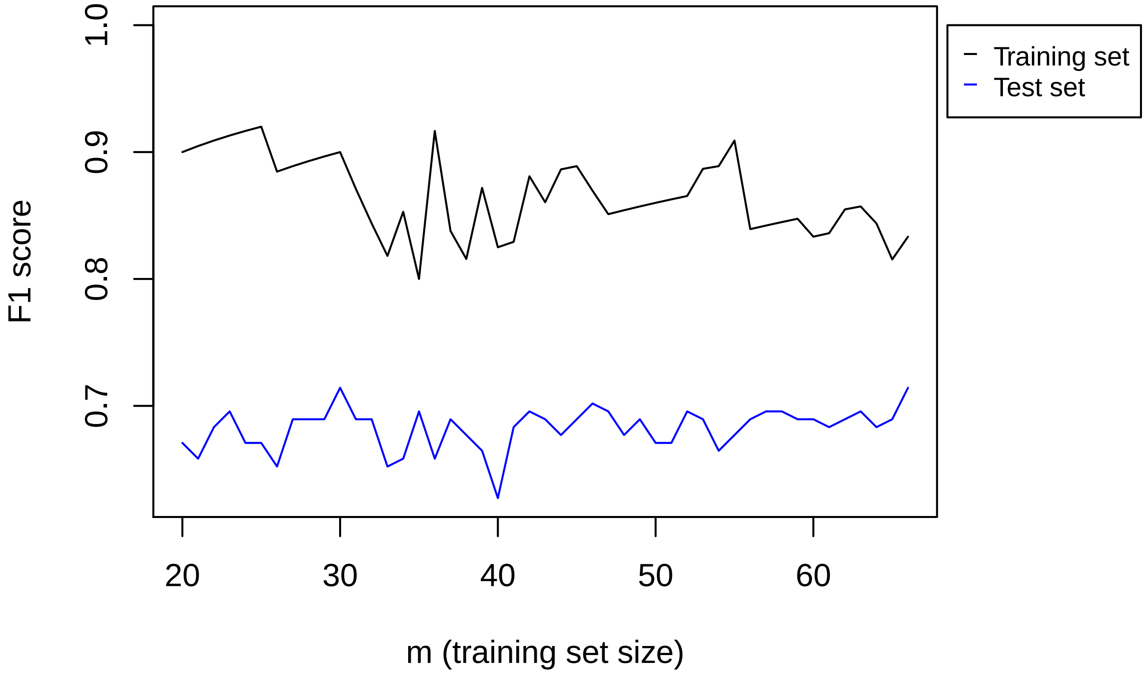
```
## [1] "The optimized theta values for lambda= 0.909 are:"
```

```
colnames(thetaMat) <- c("Intercept", colnames(trainSet[,inputVars]))
thetaMat[best.lambda.idx,] %>% round(3) %>% print()
```

```
##        Intercept ERpos_5pseg3_sig ERpos_CM_16p_sig ERpos_CM_16q_sig
##            3.351            0.294            0.409            0.686
##  ERneg_CM_2p_sig  ERneg_CM_4p_sig  ERneg_CM_4q_sig  ERneg_CM_5q_sig
##           -0.788           -1.766           -1.138           -0.458
##  ERneg_CM_6p_sig ERneg_CM_10q_sig ERneg_CM_14q_sig
##           -0.771           -0.389           -0.432
```

Learning curve: increasing number of samples

```
curve <- learningCurve(initial_theta, best.lambda, X, y, Xval, yval, metric="F1")
```


F1 score
1.0
0.9
0.8
0.7
20
30
40
50
60
m (training set size)
Training set
Test set