Lab3 block2 732A95

Anton Persson antpe404

16 december 2016

Assignment 1 High dimensional methods

The first assignment is about different methods to deal with wide data.

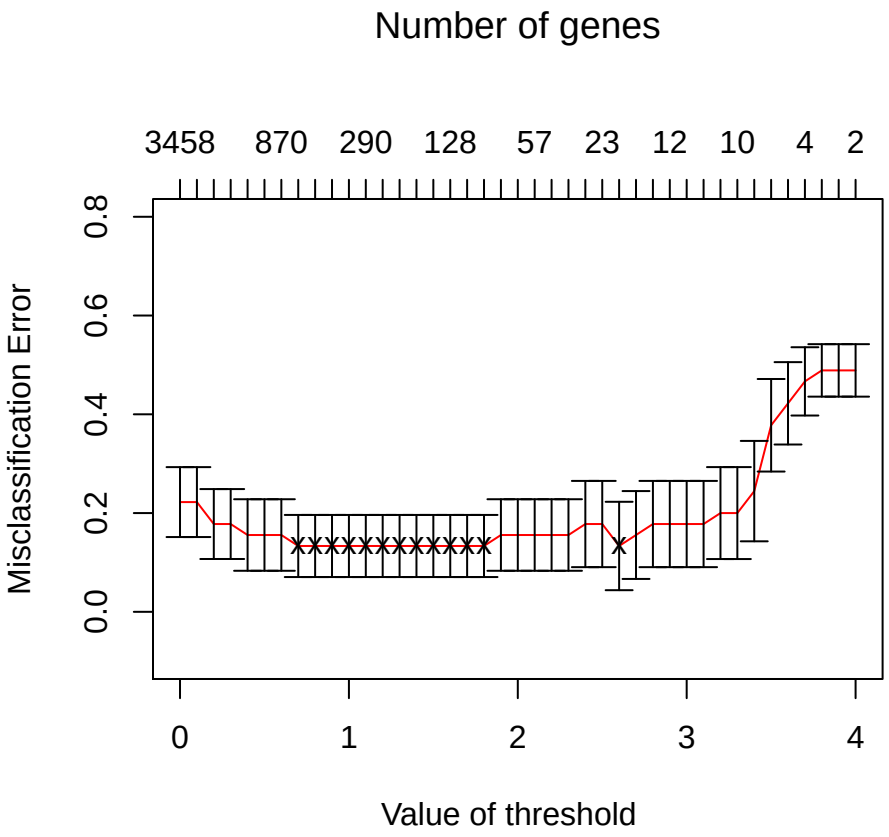
Assignment 1.1

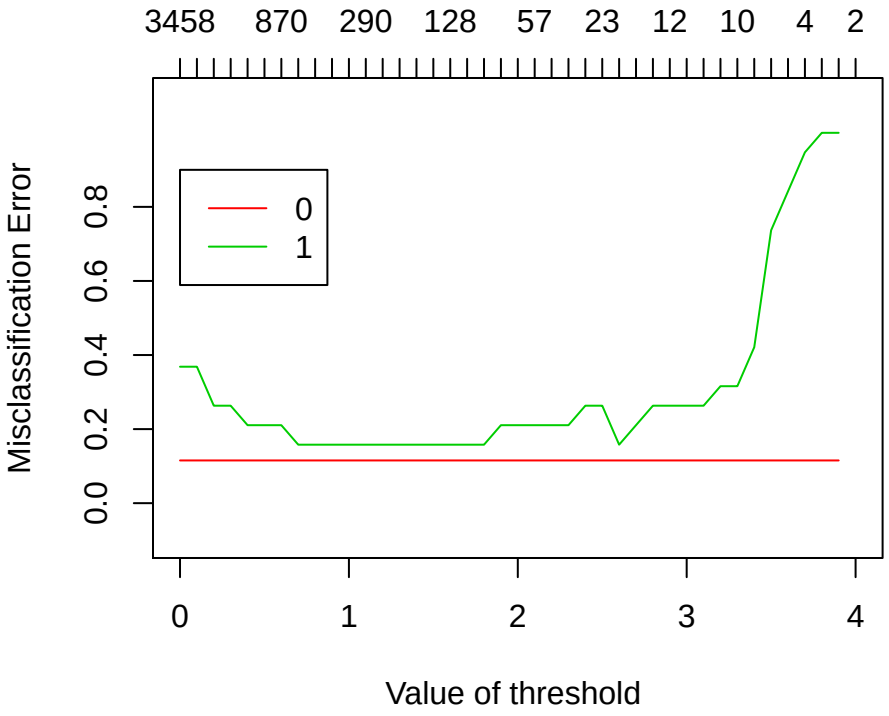
I divided the data into training and test according to the instruction. The nearest shrunken centroid classification was done and the required results are presented below.

```
## Call:
## pamr.cv(fit = model, data = mydata)
##    threshold nonzero errors
## 1  0.0       3458    10
## 2  0.1       3428    10
## 3  0.2       3110    8
## 4  0.3       3042    8
## 5  0.4       3025    7
## 6  0.5       1977    7
## 7  0.6        870    7
## 8  0.7        850    6
## 9  0.8        673    6
## 10 0.9        671    6
## 11 1.0        295    6
## 12 1.1        290    6
## 13 1.2        269    6
## 14 1.3        234    6
## 15 1.4        154    6
## 16 1.5        151    6
## 17 1.6        128    6
## 18 1.7        100    6
## 19 1.8         97    6
## 20 1.9         73    7
## 21 2.0         64    7
## 22 2.1         57    7
## 23 2.2         42    7
## 24 2.3         37    7
## 25 2.4         35    8
## 26 2.5         23    8
## 27 2.6         21    6
## 28 2.7         20    7
## 29 2.8         14    8
## 30 2.9         12    8
## 31 3.0         11    8
## 32 3.1         10    8
## 33 3.2         10    9
```

```
## 34 3.3         10    9
## 35 3.4          9    11
## 36 3.5          4    17
## 37 3.6          4    19
## 38 3.7          4    21
## 39 3.8          4    22
## 40 3.9          3    22
## 41 4.0          2    22
```

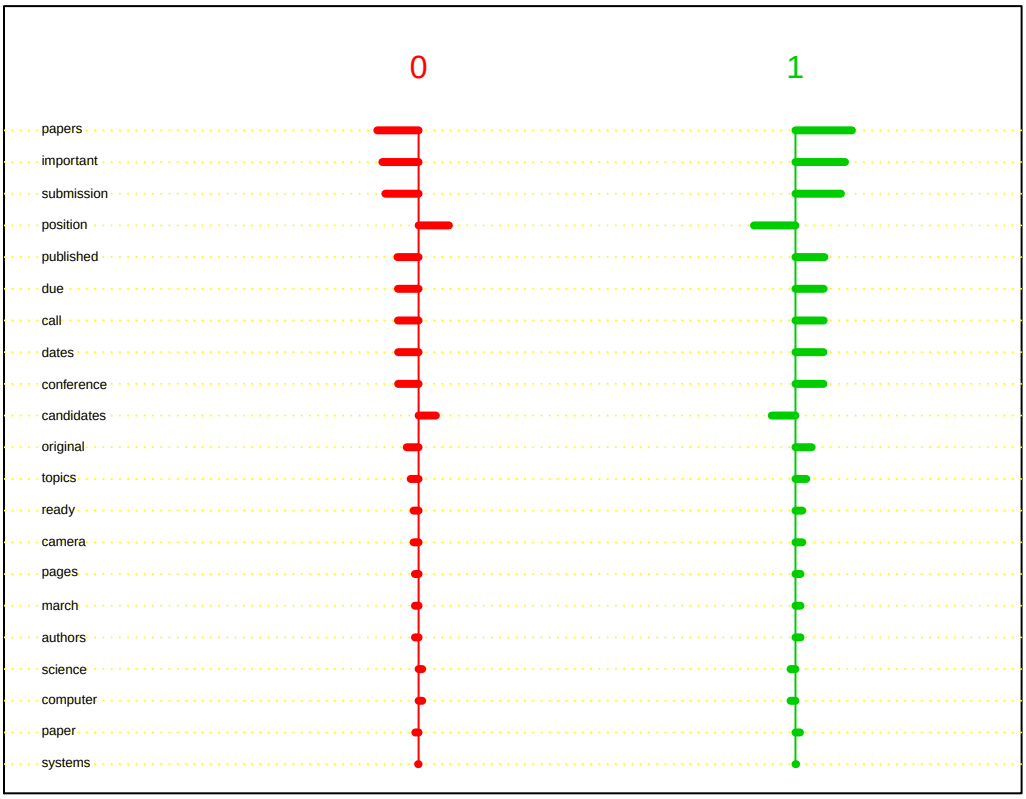
From the table above, I conclude that a threshold between 0.7-1.8 or 2.6 generates the lowest error. I choose 2.6 as threshold because of simplicity. This will generate 21 selected features, notated as *nonzero* in the output above. The figure below visualizes the table above. The plot on top in the figure below says that above mentioned thresholds minimizes the error.


Number of genes
3458 870 290 128 57 23 12 10 4 2
Misclassification Error
0.0 0.2 0.4 0.6 0.8
0 1 2 3 4
Value of threshold


3458 870 290 128 57 23 12 10 4 2
Misclassification Error
0.0 0.2 0.4 0.6 0.8
0
1
0 1 2 3 4
Value of threshold

The ten most important features for the model with 2.6 as threshold is visualized in a cendriod plot below.

```
## 1
```



The deafult plot in pamr doesn't have a brilliant layout, why it's kind of hard to read which variables are actually chosen. For clarity I decided to list them in a more proper way. The ten most contributing features are thus listed below.

```
## papers
## important
## submission
## position
## published
## call
## due
## conference
## dates
## candidates
```

I'd say that the words are reasonable. I can see why you'd mention words like *paper*, *submission*, *candidates*, *published*, *dates* and *conference* in a mail about conferences. Finally, the test error is presented below. MENTION STUFF ABOUT 0/1, positive 1 means occurance tenderar att explaina 1. negativt 0 means ordens icke occurance explains att mailet is 0, not about Conference.

```
## [1] 0.1052632
```

Assignment 1.2

In this assignment I'm supposed to compute the error rate and number of contributing features for two more methods, elastic net and support vector machine. I start off with elastic net.

Assignment 1.2a

The instructions defines the type of response and value of α for me. I do use the function cv.glmnet to decide penalty by cross validation. The selected penalty, the number of features and test error rate is presented below.

```
## $selected_penalty
##            deviance
## "Binomial Deviance"
##
## $test_error_rate_elastic
## [1] 0.1578947
##
## $number_of_features_elastic
## [1] 12
```

Assignment 1.2b

The test error and number of contributing features for a SVM with *vanilladot* kernel are presented below.

```
##  Setting default kernel parameters

## $test_error_rate_svm
## [1] 0.05263158
##
## $number_of_features_svm
## [1] 4702
```

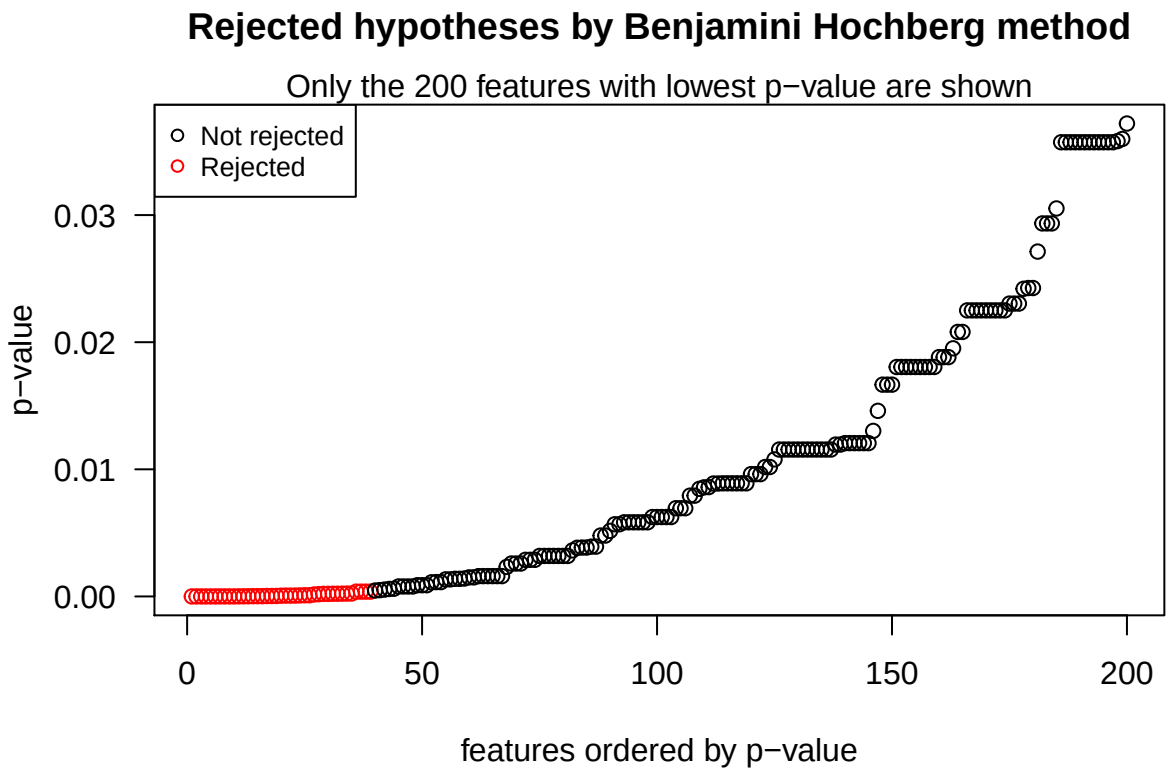
Finally I compare the three methods used above, nearest shrunken centroid method, elastic net and support vector machine, by arranging a table with all the results. Watch it below!

```
##                         NSC Elastic       SVM
## Number of features 21.0000 12.0000 4702.0000
## test_error_rate     0.1053  0.1579    0.0526
```

From the table I conclude that I'd probably choose the NSC method. NSC has one more missclassification than the SVM, but reduces the number of features from 4702 to 21. The SVM method has the lowest error rate, but uses all features. SVM doesn't reduce the number of features, but try to reduce the number observations as much as possible. In this case it only reduces the number of rows from 45 to 44 though, since I have 45 rows in my training set and 44 support vectors in my ksvm-modeln. That's not much of a reduction.

Assignment 1.3

In this assignment I implemented the Benjamini Hochberg method on the complete original data, not the training set only. To begin with, the result is visualized by the plot below.



Notice that in the plot above, only the first 200 features of the ordered data frame are plotted. That means that the features visualized in the plot only corresponds to the about 4 % of all features in the data. I did this because of off the 4702 features tested, only 39 gets rejected. Those 39 rejected features are thus the features that are most useful for describing the feature *Conference*.

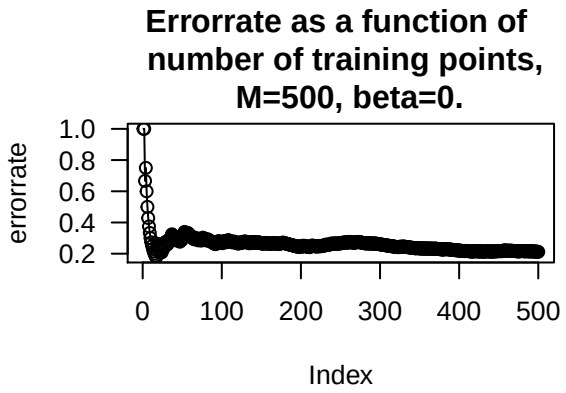
The 39 features that were rejected, i.e. gets chosen, are shown below. Notice that the words that have occured as important features in earlier methods are among the rejected features for the Benjamini Hochberg method as well.

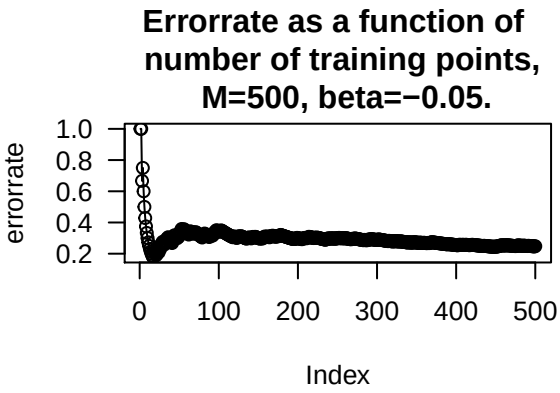
```
## papers
## submission
## position
## published
## important
## call
## conference
## candidates
## dates
## paper
## topics
## limited
## candidate
```

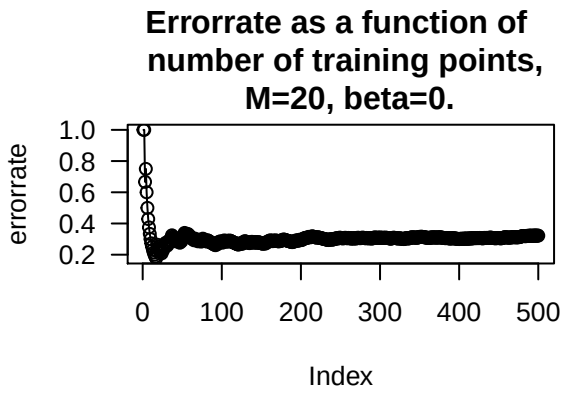
```
## camera
## ready
## authors
## phd
## projects
## org
## chairs
## due
## original
## notification
## salary
## record
## skills
## held
## team
## pages
## workshop
## committee
## proceedings
## apply
## strong
## international
## degree
## excellent
## post
## presented
```

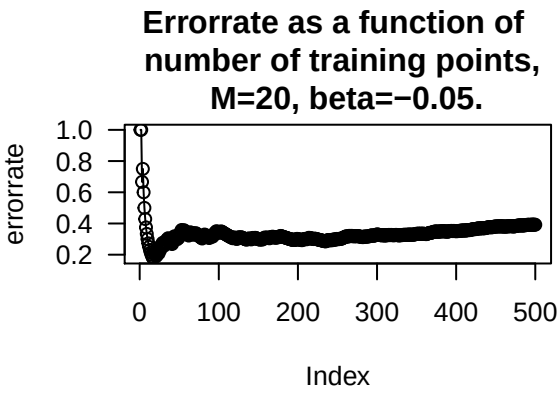
Assignment 2 Online learning

I used the pseudo code in the slides to implement the budget online SVM. I used parts of the template in the instruction as well. The four plots of the error rates as a function of the number of training points are shown below.









Klura lite paa visualiseringen. Antingen en plot dar alla visas i GG. Eller en rubrik, övergripande, samt bara M och beta i de fyra plottarna.

Have to think about whether beta -0.05 gives a smoother result than 0.

The call with M=20, beta=0 is the slowest because it generates the most computation. The higher the β is the more iterations are true in step 5 of the pseudo code, thus the more computation needs to be done. That also applies for M=20 compared to M=500. When M=500, the $|S|$ never gets bigger than M when I run 500 training points. That means that the exclusion of the least important support vector in step eight in the pseodo code is never computed.