

**ATHENS UNIVERSITY OF ECONOMICS & BUSINESS**

MSc in Data Science

**Statistics for Big Data**

**Final Project – Supervised PCA**

Kalderon Leon

Sarri Georgia

Prandeka Maria

**1. Introduction**

Classification of breast cancer patients into different risk classes is very important in clinical applications. The cause of breast cancer is not common to every patient. It is estimated to be highly correlated with a combination of gene expression data.

In this project, we are dealing with data referring to 78 patients and part of their gene profile, around 25.000 genes. In particular, the aim of our project is to identify among the genes, those that are related the most to the time of survival.

From the biological aspect, only a small portion of genes contributes to the patients’ survival time. A substantial challenge in this context comes from the fact that the number of genes is usually much larger than the number of patients. Furthermore, it is very difficult to select the most significant of them, as these may depend on each other. Because of the large number of genes, it is easy to find genes that are significant for the specific sample of patients, but not significant for others, leading to misleading rules. Thus, a crucial step is the dimensional reduction from the gene expression profiles.

In this exercise, we explore the use of the method called Supervised   
Principal Component Analysis (SPCA). Specifically, this is a technique that finds underlying variables (known as principal components) that best differentiates our genes dataset. In particular, principal components are dimensions along which the genes are most spread out. Moreover, these dimensions are also related to the survival time of the patients. A principal component can be expressed by one or more existing genes. Therefore, instead of analyzing all genes dataset, we combine highly correlated genes, that are also associated with the survival time, leaving just a group of them to consider. Finally, from this group we derive only the genes that contribute mostly to the survival.

In the next section we explore our dataset and describe the pre-processing and filtering of the dataset. In Section 3 we describe the model that we use in order to derive the most important genes in relation to the patients’ survival time. In Section 4 we mention some conclusions to our analysis.

**2. Data Preprocessing**

**2.1 Dataset Description**

Our original dataset consists of observations taken from 78 patients having breast cancer with gene expression measurements for 24.481 genes. The outcome was survival time measured in months and ranging from 3 months to 13,4 years.

**2.2 Handling Missing Information**

**2.2.1 Missing Gene information per patient**

For all our patients missing information was observed regarding a certain amount of genes. The average amount of missing data per patient was around 300 genes with two exceptions where for one patient we had more than 10000 missing observations and for the other approximately 3000 [Figure 1.1].

*Figure 2.1: Missing Information*



To proceed with our analysis the patient with the 10,000 missing observations was removed from the dataset and the rest were filled using the imputation process.

**2.2.2 Missing Gene Information for all patients**

Throughout the dataset 293 genes were observed that had no information across the patients, those genes were removed from the dataset.

**2.2.3 Imputing Missing Data for Gene Expression Arrays**

For the missing information of genes per patient the imputing method was used to fill the gaps so as to not omit information for useful genes by removing completely the ones having at least one missing observation.

In general, there are about 3200 genes that have 1-10 missing observations (NaNs). The process that we followed in order to fill them was proposed on the paper “Imputing Missing Data for Gene Expression Arrays”(TrevorHastie, 1999) which is written from the same researchers that have written the paper for Supervised Principle Component Analysis. This paper concerns specifically microarrays gene imputation. From the three different methods described in the paper, k-Nearest-Neighbor imputation had the best results in their experiments thus we implemented this method in our analysis.

The underline process of this method is:

1. Compute the Euclidean distance between xi and all the genes in X, using only the non-missing xi. Identify the K closest.
2. Impute the missing coordinates of xi by averaging the corresponding coordinates of the K closest.

For our analysis we chose to impute the missing genes with a k=5 Nearest-Neighbor imputation Algorithm.

* + 1. **Filtering the Dataset**

The main issue regarding the microarray dataset is the large number of features comparing to the observations. In order to proceed with the analysis a form of pre-filtering is necessary.

In order to pre-filter the dataset, a correlation test was used to keep the genes that are the most correlated with the months of survival. For the correlation test we used the Spearman Test since it is a non-parametric type of test for correlation between a categorical and continuous variables and it is more lenient regarding the model’s assumptions.

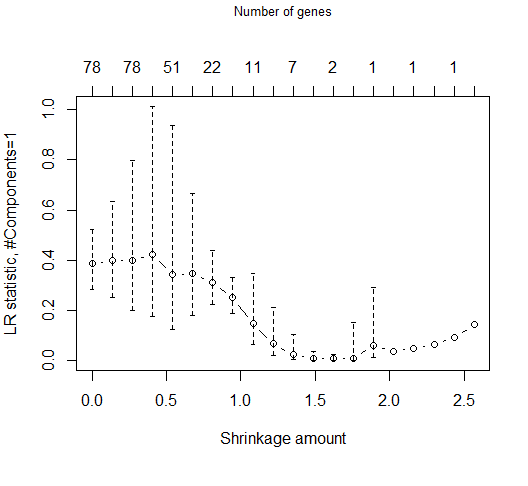
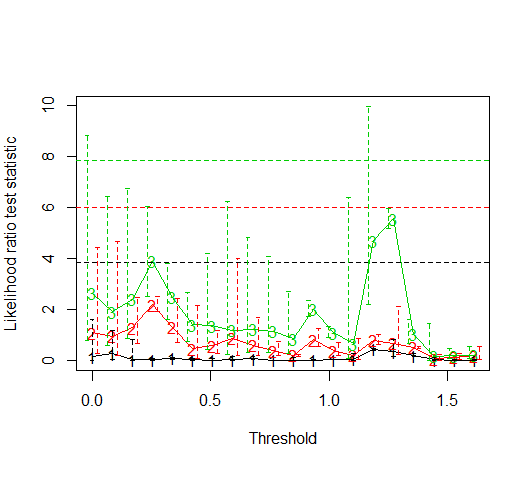
Two different approaches were tested for the pre-filtering. One was to run a correlation test one time and keep only the coefficients that were above a 5% significance level [Figure 2.2] and the second was to split the data randomly in two halves and check significance in every sample (i.e. for each gene test twice) at significance level of 5% [Figure 2.3]. Finally, we choose the genes that are significant in both tests [Table 2.1].

*Table 2.1*

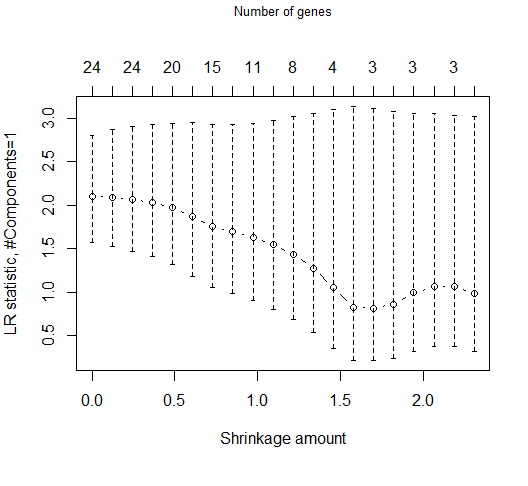
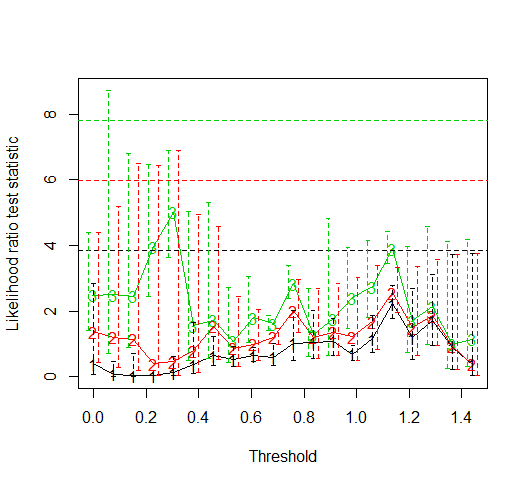
|  |  |  |
| --- | --- | --- |
|  | 1 Step Correlation | 2 Step Correlation |
| Threshold | 0.05 | 0.05 |
| Number of Remaining Genes | 3456 | 897 |

Note that from the 897 genes remaining after the 2 Step Correlation the 821 are in the larger set of 3546.

*Figure 2.2 - 1 STEP CORRELATION*



*Figure 2.3 - 2 STEP CORRELATION*



We observe that after applying the SPCA to the two different filtered Gene Samples, the final results differ (only one gene in common). Note that in the following analysis, we are going to use the results from the 1 step correlation as the input dataset to the SPCA model. Given the nature and the sensitivity of the problem examined, we decided to follow the approach of the 1 step correlation, as it results in a larger pool of genes.

**3. Supervised Principal Component Analysis**

In order to identify the genes that are correlated the most with the time of survival, we used the Supervised Principal Component Analysis.

Supervised principal components is a generalization of principal components regression. The first (or first few) principal components are the linear combinations of the features that capture the directions of largest variation in a dataset. But these directions may or may not be related to an outcome variable of interest. To find linear combinations that are related to an outcome variable, we compute univariate scores for each gene and then retain only those features whose score exceeds a threshold. A principal components analysis is carried out using only the data from these selected features.

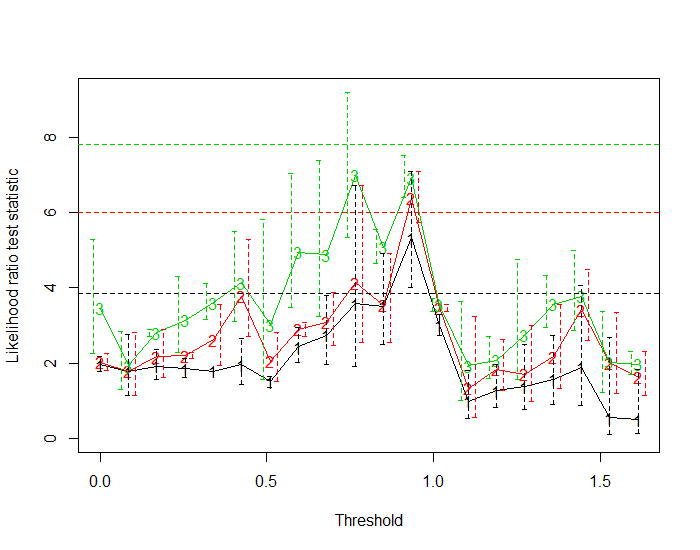
Finally, these "supervised principal components" are used in a regression model to predict the outcome. To summarize, the steps are:

1. Compute (univariate) regression coefficients for each gene.
2. Form a reduced data matrix consisting of only those genes whose univariate coefficient exceeds a threshold theta in absolute value (theta is estimated by cross-validation)
3. Compute the first (or first few) principal components of the reduced data matrix
4. Keep the genes most correlated to the principal components.

**3.1 Description of the model**

In order to obtain principal components related to survival we run univariate survival regression of each gene with the time (in months) that the patient survived. Then we train our model and derive the log-likelihood ratio test statistic to get the first three principal components [Figure3.1].

*Figure 3.1*



The Likelihood-Ratio helps us choose the “best” model between two or more nested models i.e. the one nested model is a special case of the other. For example, model1 could be a survival regression with predictors gene1, gene2, gene3 and model2 another survival regression with gene2, gene3 as predictors.

In our case, for different thresholds on the coefficient we get different genes (the biggest the threshold the less the genes). In Figure 3.1, the x axis shows the different thresholds and the y axis shows the corresponding Likelihood-Ratio of the model at which gene's absolute coefficients are larger than the threshold.

The maximum Likelihood Ratio of the 1rst Principal component holds for threshold equal to 0.92. For this threshold we reduce the selected genes to 258 from 3546.

In the next step, we use these 258 genes to calculate the eigengenes that we finally regress on the survival time. By this process, we derive the statistical significant principal component(s) [Table 3.1]. As we observe from the table below, the 1st principal component is by far the most statistically significant component.

*Table 3.1*

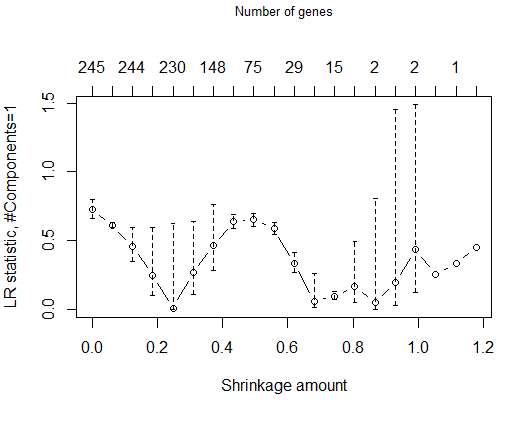
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **coef** | **exp.coef.** | **se.coef.** | **z** | **Pr(>|z|)** |
| **score.1** | 25.5693543 | 1.2724E+11 | 3.56641198 | 7.16948978 | 7.53E-13 |
| **score.2** | 4.05406239 | 57.631102 | 1.62590634 | 2.49341693 | 0.012652019 |
| **score.3** | 1.89114067 | 6.62692348 | 1.92544068 | 0.98218589 | 0.326008277 |

Although, we found the component mostly related to the survival time, we do not know the specific genes compose this axis. In order to obtain the most significant genes which contribute most to the prediction of survival time, we need to define the importance score i.e. the inner product between each gene and the principal component.

Finally, for different shrinkage amount on the importance scores we get different number of genes (the biggest the shrinkage amount the less the genes) [Figure 3.2]. The x-lower axis shows the different shrinkage amounts and the x-upper axis the number of genes. The y axis shows the corresponding Likelihood-Ratio of the model at which gene's absolute important scores are larger than the shrinkage amount.

The maximum Likelihood Ratio holds for shrinkage amount equal to ~0.5 that corresponds to 75 genes [Table 3.2].

*Figure 3.2*



*Table 3.2 – Most important genes*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Importance.score** |  | **Name** | **Importance.score** |
| Contig53968\_RC | -1.239 |  | Contig45022\_RC | 0.576 |
| NM\_000918 | 1.084 |  | AJ223353 | -0.572 |
| Contig52976\_RC | 0.847 |  | NM\_000951 | 0.568 |
| Contig31242\_RC | -0.844 |  | Z24724 | 0.568 |
| Contig44758\_RC | -0.833 |  | NM\_006371 | 0.564 |
| NM\_006314 | 0.82 |  | NM\_018615 | -0.563 |
| NM\_002044 | 0.815 |  | Contig38448\_RC | 0.56 |
| Contig55771\_RC | 0.805 |  | Contig43673\_RC | 0.559 |
| NM\_014347 | 0.79 |  | NM\_001249 | -0.559 |
| NM\_005245 | -0.783 |  | NM\_002762 | 0.558 |
| Contig3607\_RC | -0.777 |  | Contig31378\_RC | 0.556 |
| NM\_018379 | 0.771 |  | Contig25944\_RC | 0.554 |
| NM\_005209 | 0.75 |  | NM\_015929 | -0.552 |
| NM\_004684 | -0.745 |  | NM\_004228 | 0.552 |
| Contig55004 | -0.744 |  | NM\_000856 | -0.541 |
| NM\_013409 | -0.717 |  | Contig39989\_RC | 0.54 |
| AL049963 | -0.716 |  | NM\_013260 | 0.537 |
| Contig11065\_RC | -0.709 |  | NM\_004904 | 0.535 |
| NM\_000559 | 0.692 |  | Contig51885\_RC | 0.534 |
| NM\_006198 | 0.672 |  | Contig43169\_RC | 0.533 |
| NM\_015981 | -0.665 |  | NM\_004791 | 0.532 |
| Contig54379\_RC | 0.664 |  | Contig21269\_RC | 0.529 |
| Contig36739\_RC | -0.659 |  | Contig1239\_RC | -0.529 |
| NM\_005238 | 0.64 |  | Contig16242\_RC | -0.527 |
| Contig67182\_RC | 0.64 |  | NM\_018321 | 0.526 |
| Contig40496\_RC | 0.635 |  | Contig33814\_RC | 0.525 |
| NM\_001569 | -0.629 |  | Contig25090\_RC | -0.522 |
| X93921 | -0.626 |  | U41387 | -0.521 |
| Contig49670\_RC | 0.624 |  | AF131828 | 0.52 |
| NM\_016581 | -0.597 |  | AL080059 | 0.517 |
| NM\_012261 | 0.594 |  | NM\_001617 | 0.511 |
| NM\_002292 | -0.593 |  | Contig57447\_RC | 0.507 |
| NM\_014584 | 0.592 |  | NM\_004286 | 0.505 |
| NM\_002254 | -0.59 |  | D43950 | 0.503 |
| Contig13657 | 0.588 |  | AB014586 | -0.502 |
| Contig55079\_RC | -0.587 |  | NM\_020197 | 0.502 |
| NM\_015878 | 0.579 |  | U79457 | -0.5 |
| AF176012 | -0.577 |  |  |  |

**4. Conclusions**

Supervised principal components analysis approaches the substantial challenge of having a much larger number of genes than the number of patients. With this method we managed to combine highly correlated genes, that are also associated with the survival time, leaving just a group of them to consider. Finally, from this group we derive only the genes that are mostly correlated to the survival time.

Throughout the analysis we observed a large variability in our final gene selection each time we were tweaking with the thresholds of the SPCA or by changing the pre-processing dataset.

Generally, SPCA is a compelling tool to use due to its simplicity but we did not manage to get a stable solution when using it. This may stem from the fact that the number of observations is very small (only 75 patients) comparing to the features (25.000 genes).

References

1. T. Hastie, R. Tibshirani, G. Sherlock, M. Eisen, P. Brown, D. Botstein. Imputing Missing Data for Gene Expression Arrays. Stanford University. 1999.
2. E. Bair, T. Hastie, D. Paul, R. Tibshirani. Prediction by supervised principal components. Stanford University. 2004.
3. H.M. Bøvelstad, S. Nygård, H.L. Størvold, M. Aldrin, Ø. Borgan, A. Frigessi, O.C. Lingjærde. Predicting survival from microarray data—a comparative study. Bioinformatics, Volume 23, Issue 16, 15 August 2007, Pages 2080–2087.
4. M. Farhadian, H. Mahjub, J. Poorolajal, A. Moghimbeigi, M. Mansoorizadeh. Predicting 5-Year Survival Status of Patients with Breast Cancer based on Supervised Wavelet Method. Elsevier. 2014.
5. Superpc for R: Tutorial (<http://statweb.stanford.edu/~tibs/superpc/tutorial.html>)