ROBPCA: A New Approach to Robust Principal Component Analysis

Mathematics for Machine Learning - Project 3 Report

Filipe Martins - 50710 André Roque - 86694

Instituto Superior Técnico - Universidade de Lisboa

Abstract

Classical Principal Component Analysis is based on the empirical covariance matrix of the data and hence is highly sensitive to outlying observations. The Robust Principal Component Analysis is developed with the intent of generating the Principal Components which fits best the data and minimizing the deviation effect of outliers when present in the data. The Robust Principal Component Analysis combines two robust approaches that have been developed to date, Projection Pursuit and a robust scatter matrix for robust covariance estimation. Here, we present a replication simulation with a multivariate Normal contamination model and perform a comparison analysis between classical and robust methods to the White Wine dataset.

Keywords: Principal Component Analysis, Robust Principal Component Analysis and Outliers

1 Introduction

A very popular statistical method for data reduction is Principal Component Analysis (PCA), it is thus widely used in the analysis of high-dimensional datasets. PCA is then often one of the first stages of the data analysis, and is therefore a very important method.

In real-world datasets it often happens that some observations behave differently from the majority of data. Such data points are designated by outliers. These peculiar observations may spoil the results of an analysis of data, in the event that they arise from errors, or they may also contain valuable information. It is well known that classical variance and classical covariance are very sensitive to anomalous observations. Therefore, the development of methods for detection and classification of outliers becomes crucial.

1.1 Principal Component Analysis

PCA tries to explain the covariance structure of data by means of a small number of components. These components are called Principal Components (PC) and form an orthonormal coordinate system of axes (change of base) to represent the data. The first PC corresponds to a linear combination of the original variables that explains the most variance of the data. The second PC explains the most variance in what is left once the effect of the first component is removed, is then orthogonal to the previous PC, and continuing in this way produces all of the principal components, until all the variance is explained. These PC's correspond to the eigenvectors of the empirical covariance matrix.

As previously mentioned, both the variance and the covariance estimators are very sensitive to outliers, consequently, the first PC's are often attracted toward outlying points, and may not capture the variance inherent in regular observations. This implies that data reduction based on Classical PCA (CPCA) becomes unreliable if outliers are present in the data. Therefore, the aim of Robust Principal Component Analysis (ROBPCA) is to obtain principal components that are not influenced much by outliers.

1.2 Robust PCA

The proposed ROBPCA method in [1] combines ideas of both Projection Pursuit and robust covariance estimation. We presente below a very brief explanation of the ROBPCA method. Let $\boldsymbol{X}$ be the data matrix, with dimensions $n \times p$, where n corresponds to the number of observations and p to the number of independent variables, which in turn corresponds to the dimension of the data points. The ROBPCA method then proceeds in three major steps. First, the data is preprocessed such that the transformed data is lying in a subspace whose dimension is at most $n-1$. Second, a preliminary covariance matrix $\boldsymbol{S_0}$ is generated and used for selecting k components which span a k-dimensional subspace that fits the data well. Then the data points are projected on this subspace where their location and scatter matrix are robustly estimated, and from which its k nonzero eigenvalues $l_1, ..., l_k$ are calculated. The associated eigenvectors correspond then to the k robust PC's. These eigenvectors form the $p \times k$ matrix $\boldsymbol{P_{p,k}}$, with orthogonal columns. In the original space of the data points, of dimension p, these k vectors span a k-dimensional subspace. The mean estimate, $\hat{\mu}$ is called the robust center. The scores correspond to the entries of the $n \times k$ matrix

$$\boldsymbol{T_{n,k}(X)} = (\boldsymbol{X} - \boldsymbol{1}\hat{\boldsymbol{\mu}}^\top)\boldsymbol{P_{p,k}}, \tag{1}$$

where $\boldsymbol{1}$ corresponds to a column n-vector with all components equal to 1. Furthermore, the k robust PC's generate a $p \times p$ robust scatter matrix $\boldsymbol{S}$ of rank k and given by

$$\boldsymbol{S} = \boldsymbol{P_{p,k}}\boldsymbol{L_{k,k}}\boldsymbol{P_{p,k}^\top}, \tag{2}$$

where $\boldsymbol{L_{k,k}}$ is a diagonal matrix with the eigenvalues $l_1, ..., l_k$. The ROBPCA method is location and orthogonal equivariant, such as CPCA, meaning that when a shift and or an orthogonal transformation is applied to the data, the robust center is shifted and the loadings are rotate, accordingly. So that, the scores do not change under this transformations. Now, let $\boldsymbol{A_{p,p}}$ be an orthogonal transformation thus, A is of full rank and $\boldsymbol{A}^\top = \boldsymbol{A^1}$, and let $\hat{\boldsymbol{\mu}}_{\boldsymbol{X}}$ and $\boldsymbol{P_{p,k}}$ designate the ROBPCA center and loading matrix for the original data matrix $\boldsymbol{X}$. Then, considering the application of $\boldsymbol{A}$ to the data plus a shift $\boldsymbol{v}$: $\boldsymbol{XA}^\top + \boldsymbol{1v}^\top$, we obtain the following ROBPCA center and loadings for the transformed data: $\boldsymbol{X}\hat{\boldsymbol{\mu}}_{\boldsymbol{X}} + \boldsymbol{v}$ and $\boldsymbol{AP_{p,k}}$, respectively. Consequently, the scores do not change under these transformations, since:

$$\begin{aligned}
&\boldsymbol{T_{n,k}}(\boldsymbol{XA}^\top + \boldsymbol{1v}^\top) = \\
&= (\boldsymbol{XA}^\top + \boldsymbol{1v}^\top - \boldsymbol{1}(\boldsymbol{X}\hat{\boldsymbol{\mu}}_{\boldsymbol{X}} + \boldsymbol{v})^\top)\boldsymbol{AP_{p,k}} \\
&= (\boldsymbol{X} - \boldsymbol{1}\hat{\boldsymbol{\mu}}_{\boldsymbol{X}})\boldsymbol{P_{p,k}} = \boldsymbol{T_{n,k}}(\boldsymbol{X}).
\end{aligned}$$

The detailed description of the ROBPCA algorithm can be seen in the Appendix of [1].

1.3 Outlier detection

Outliers may be caused by errors, or could represent observations that have been recorded under exceptional circumstances, or even could belong to a different data population. Therefore, these peculiar observations may spoil the results of an analysis of data or they may also contain valuable information. Therefore, the development of methods for detection and classification of outliers becomes crucial.

Given a set of observations and the respective associated PCA subspace, we may classify each observation in one of the following types:

1. **Regular observations** - a homogeneous group of observations that is close to the PCA subspace

2. **Good leverage points** - points that lie close to the PCA subspace

3. **Orthogonal outliers** - points with a large orthogonal distance to the PCA subspace, but that are invisible when we consider their projection in the PCA subspace

4. **Bad leverage points** - points that simultaneously have a large orthogonal distance to the PCA subspace, but their projection on the PCA subspace is distant from the remaining observations projections on that space.

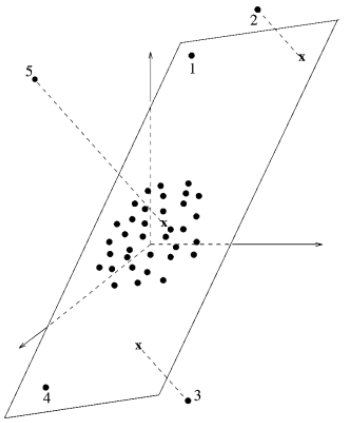

Figure 1: Illustration of the 4 types of observations. The plane represents the PCA subspace, the points around center of the axis are regular points, points 1 and 4 are good leverage points, point 5 is a orthogonal outlier and points 2 and 3 are bad leverage points.

In order to identify which kind of point is each observation, two distances are used:

- The *Robust Score Distance SD*

$$SD_i = \sqrt{\sum_{j=1}^{k} \frac{t_{ij}^2}{l_j}} \tag{3}$$

- The *Orthogonal Distance OD*

$$OD_i = \|\mathbf{x}_i - \hat{\mu} - \mathbf{P}_{p,k}\mathbf{t}_i'\| \tag{4}$$

where t_{ij} are obtained from 2. By assuming normality on the distribution of these scores, the Mahalanobis distance assumes a χ^2 distribution. Defining the **false alarm rate** as the probability of an observation being classified as an outlier given that it is a regular observation, then the expressions for the cutoffs values for the SD and the OD are the following:

$$C_{SD} = (\chi^2_{(k)}(1 - \alpha_{SD}))^{\frac{1}{2}} \tag{5}$$

$$C_{OD} = (\mu + \xi_{1-\alpha_{OD}}\sigma) \tag{6}$$

where $\xi_{1-\alpha_{OD}}$ is the quantile of the $N(0,1)$, and μ and σ and the location and scale estimators of the orthogonal distances.

2 Simulation Replication

We replicate one of the simulations in [1] to compare the robustness of ROBPCA method with CPCA method. This is the simulation of the multivariate Normal distribution case. We have two cases: one low-dimensional case with $n = 100$ observations and $p = 4$ independent variables, which we designate by *case 1*, and one high-dimensional case with $n = 50$ observations and $p = 100$ independent variables, which we designate by *case 2*. We generate 1000 samples of size n in p dimensions from the contamination model:

$$(1-\epsilon)N_p(\mathbf{0}, \mathbf{\Sigma}) + \epsilon N_p(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{\Sigma}}), \tag{7}$$

where ϵ is the contamination factor and we consider $\epsilon = 0$, corresponding to no contamination, $\epsilon = 0.10$ and $\epsilon = 0.20$, corresponding to 10% and 20% of contamination, respectively. We have the following setting of parameters:

- *case 1* (low-dim): $n = 100$, $p = 4$, $\mathbf{\Sigma} = \text{diag}(8, 4, 2, 1)$, $\epsilon = 0, 0.10, 0.20$, $\tilde{\boldsymbol{\mu}} = (0, 0, 0, f_1)$, with $f_1 = 6, 10, 14, 18$ and $\tilde{\mathbf{\Sigma}} = \mathbf{\Sigma}/f_2$, with $f_2 = 1, 15$. Given that $\sum_{i=1}^{3}\lambda_i / \sum_{i=1}^{4}\lambda_i \approx 93.33\%$, we set the number of PC's to $k = 3$. We obtain 17 combinations;

- *case 2* (high-dim): $n = 50$, $p = 100$, $\mathbf{\Sigma} = \text{diag}(17, 13.5, 8, 3, 1, 0.095, 0.094, ..., 0.001)$, $\epsilon = 0, 0.10, 0.20$, $\tilde{\boldsymbol{\mu}} = (0, 0, 0, 0, 0, f_1, 0, ..., 0)$, with $f_1 = 6, 10, 14, 18$ and $\tilde{\mathbf{\Sigma}} = \mathbf{\Sigma}/f_2$, with $f_2 = 1, 15$. Given that $\sum_{i=1}^{5}\lambda_i / \sum_{i=1}^{100}\lambda_i \approx 90.31\%$, we set the number of PC's to $k = 5$. We obtain 17 combinations.

We consider three measures to assess the quality of computational results. We considered the maximal angle between,*maxsub*, $\boldsymbol{E_k}$ and the estimated PCA subspace, which is spanned by the columns of $\boldsymbol{P_{p,k}}$, and where $\boldsymbol{E_k} = \text{span}\{\boldsymbol{e_1}, ..., \boldsymbol{e_k}\}$ corresponds to the subspace spanned by the first k eigenvectors of $\mathbf{\Sigma}$, being diagonal, each $\boldsymbol{e_j}$ is the j-th column of $\boldsymbol{I_{p,k}}$. We compute the mean proportion of variability that is explained by the estimated eigenvalues by using the formula below:

$$\frac{1}{1000}\sum_{l=1}^{1000}\frac{\hat{\lambda}_1^{(l)} + ... + \hat{\lambda}_k^{(l)}}{\lambda_1 + ... + \lambda_p}, \tag{8}$$

where $\hat{\lambda}_i^{(l)}$ is the estimated eigenvalue at the l-th sample and λ_i is the eigenvalue of $\mathbf{\Sigma}$ (corresponding to each diagonal entry). And we also compute the mean squared error (MSE) between the estimated and true eigenvalues, for the k largest eigenvalues, defined by the formula below:

$$\text{MSE}(\hat{\lambda}_j) = \frac{1}{1000}\sum_{i=1}^{1000}(\hat{\lambda}_j^{(l)} - \lambda_j)^2, \;\; j = 1, ..., k. \tag{9}$$

2.1 Computational Methodology

In order to perform the simulation described in the preceding subsection we considered the following steps:

1. We define the parameters presented in the previous items for *case 1* and *case 2*;
2. We determined the proportion of population total variance for $k = 3$ for $\mathbf{\Sigma}$ in *case 1* and for $k = 5$ for $\mathbf{\Sigma}$ in *case 2*;
3. Using different seeds we randomly generated 1000 samples for $N_p(\mathbf{0}, \mathbf{\Sigma})$ and $N_p(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{\Sigma}})$, for both cases considered, by making use of *mvrnorm* available in *MASS* R package [2];
4. We then generated the associated contamination models, using formula (7);
5. We defined the functions *get_results_robpca_n*100, *get_results_robpca_n*50, *get_results_cpca_n*100 and *get_results_cpca_n*50, to perform ROBPCA and CPCA for each sample, in a loop, generating a list of lists with the results for each sample. For ROBPCA we used *robpca* available in *rospca* R package [3], which implements the algorithm proposed in [1], and for CPCA, we used *prcomp* from *stats* R package [4];
6. We then applied the previous functions to obtain the results of ROBPCA and CPCA for both cases considered;
7. We defined two auxiliary functions: *get_data_for_results*, which collects the estimated eigenvalues/standard deviations and *maxsub* angles, between $\boldsymbol{E_k}$ and the subspace spanned by columns of $\boldsymbol{P_{p,k}}$ from ROBPCA/CPCA from all samples of each setting of the parameters, sums the angles and sums the eigenvalues/standard deviations with respect to each component, and *MSE_eigenvals* which calculates the MSE between the estimated eigenvalues, $\hat{\lambda}_j$, and the true eigenvalues, λ_j, from all samples of each setting of the parameters, implementing formula (9);
8. We then calculate the mean of *maxsub*, the mean proportion of explain variability, using formula (8), and the MSE's, from all samples of each setting of the parameters. And in sequence produce the tables and plots.

2.2 Results and Discussion

The results of the *maxsub* measure for the considered simulation are presented in Table 1 for no contamination data and in Figure 2 and in Figure 3 for low-dim and high-dim cases, respectively, for the contamination model 7.

The ideal *maxsub* value is 0. In Table 1, we have the values for the samples of the no contamination model, (taking $\epsilon = 0$). We observe that CPCA gets a better result in both low and high dim cases, but comparable, and for high-dim case the values increase, showing that more dimensions complexifies the computation of PC's. Concerning the contamination data, in every

situation, we observe in CPCA that *maxsub* is always very close to 1, meaning that the estimated PCA subspace has been highly influenced by the outliers in such a way that at least one PC gets almost orthogonal to $\boldsymbol{E_k}$, and when the outliers get further way from the inliers or when the factor noise increases, the estimated PCA subspace gets considerably more deviated from $\boldsymbol{E_k}$. Concerning ROBPCA we observe that the estimated PCA subspace remains close in terms of inclination to E_k and remains practically constant as the outliers get further away from the inliers, showing that ROBPCA is robust with respect to the presence of the outliers here generated.

n	p	CPCA	ROBPCA
100	4	0.093	0.130
50	100	0.215	0.249

Table 1: Results of *maxsub* values for low-dim and high-dim for CPCA and ROBPCA and no contamination, $\epsilon = 0\%$.

For the mean proportion of explained variability, the optimal values are 93.33% for low-dim case and 90.31% for high-dim case. In Table 2, in the Appendix, we present the mean proportion of explained variability for the simulation data for $f_2 = 1$, for $f_2 = 15$ the values are a little lesser and qualitatively the same with respect to each entry of the table. We observe that the mean proportions are bigger for ROBPCA than CPCA, with no contamination.

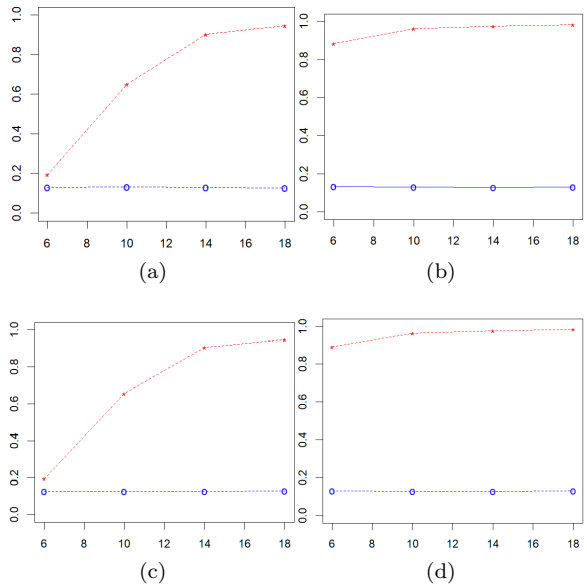

Figure 2: The *maxsub* value of the low-dim case for $f_2 = 1$, $\epsilon = 10\%$, $\epsilon = 20\%$ in (a) and (b), respectively, and $f_2 = 15$, $\epsilon = 10\%$, $\epsilon = 20\%$ in (c) and (d), respectively. In red are the values obtained with CPCA and in blue are the ones obtained from ROBPCA. In the abscissa, $f_1 = 6, 10, 14, 18$.

The mean proportion is less in high-dim than in low-dim for CPCA but the opposite occurs for ROBPCA, which is not expected, considering no contamination. In CPCA, we observe that as f_1 increases the mean proportions increase too, which is somewhat expected as the outliers are put further from the regular data this produces an increase in the eigenvalues estimations, with two situations, in contamination low-dim data, where it exceeds 100% of explainability, meaning that the eigenvalues get overestimated. In ROBPCA, the previous observation, when f_1 increases, its less evident than CPCA, and we observe much more stable results for ROBPCA, meaning that the computation of the PC's is not so much influenced by the presence of the contamination data, as in CPCA. Nonetheless, there are several values in low-dim and high-dim for CPCA and ROBPCA that are not quite as expected.

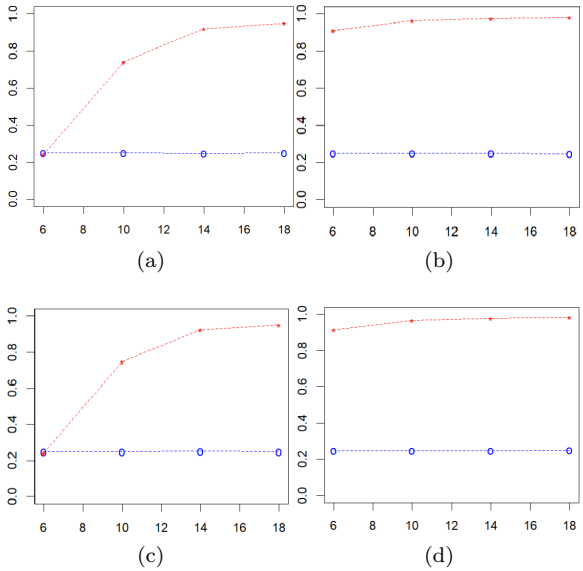

Figure 3: The *maxsub* value of the high-dim case for $f_2 = 1$, $\epsilon = 10\%$, $\epsilon = 20\%$ in (a) and (b), respectively, and $f_2 = 15$, $\epsilon = 10\%$, $\epsilon = 20\%$ in (c) and (d), respectively. In red are the values obtained with CPCA and in blue are the ones obtained from ROBPCA In the abscissa, $f_1 = 6, 10, 14, 18$.

In Figures 4 and 5 we present the results for the simulation of the MSE's ratios of the eigenvalues obtained from CPCA versus ROBPCA, for low-dim and high-dim, respectively. For both cases of dimension the results are qualitatively the same with respect to f_2, but slightly smaller ratios for $f_2 = 15$, specially for $\epsilon = 20\%$. For the low-dim case, when $\epsilon = 10\%$, we observe that the MSE's ratios of the first eigenvalue have the same order of magnitude, which is not expected and is not clear why is this result, the second eigenvalues get bigger for ROBPCA in comparison to the ones from CPCA, as f_1 increases, as the ratios get smaller, which is also not expected, and the MSE's ratios of the third eigenvalue get larger as f_1 increases, this explains *maxsub* behavior as its values approach 1. When $\epsilon = 20\%$, we observe some oscillating variation in second eigenvalue, as f_1 increases, and it gets

slightly above 1, for $f_2 = 1$, and slightly below 1, for $f_2 = 15$, when f_1 is larger.

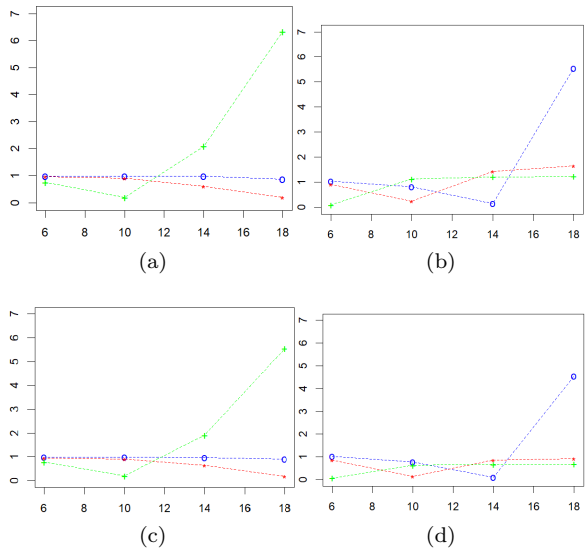

Figure 4: MSE's ratios for low-dim case for $f_2 = 1$, $\epsilon = 10\%$, $\epsilon = 20\%$ in (a) and (b), respectively, and $f_2 = 15$, $\epsilon = 10\%$, $\epsilon = 20\%$ in (c) and (d), respectively. In blue - $\hat{\lambda}_1$, in red - $\hat{\lambda}_2$ and in green - $\hat{\lambda}_3$. In the abscissa, $f_1 = 6, 10, 14, 18$.

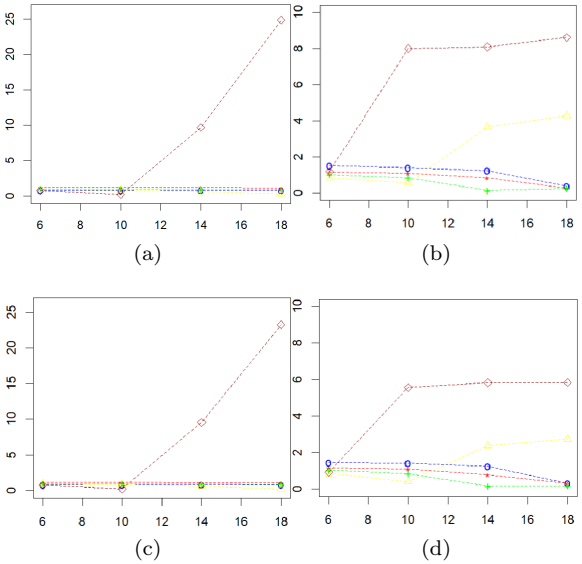

Figure 5: MSE's ratios for high-dim case for $f_2 = 1$, $\epsilon = 10\%$, $\epsilon = 20\%$ in (a) and (b), respectively, and $f_2 = 15$, $\epsilon = 10\%$, $\epsilon = 20\%$ in (c) and (d), respectively. In blue - $\hat{\lambda}_1$, in red - $\hat{\lambda}_2$, in green - $\hat{\lambda}_3$, in yellow - $\hat{\lambda}_4$ and in dark red - $\hat{\lambda}_5$. In the abscissa, $f_1 = 6, 10, 14, 18$.

The MSE's ratios of the third eigenvalue increase and stabilize at values close to 1. Whereas for the MSE's ratios of the first eigenvalue we observe a decrease until $f_1 = 14$ and a considerable increase after, meaning that the CPCA estimations get quite influenced by the further placement of the outliers from the regular data, which in turn explains the significant increase of the respective *maxsub* values. In relation to the high-dim case, when $\epsilon = 10\%$, the MSE's ratios of the first four eigenvalues show stability and are around close to 1, meaning that the respective eigenvalues have the same order of magnitude, whereas the fifth eigenvalue MSE ratio increases very significantly, meaning that the estimation from CPCA gets strongly influenced as the outliers get further from the regular data. When $\epsilon = 20\%$, the MSE's ratios of the first four eigenvalues decrease slightly as f_1 increases, which it was not expected such as in the low-dim case. meaning that the estimation from CPCA gets strongly influenced as the outliers get further from the regular data. eigenvalues decrease slightly as f_1 increases, which it was not expected such as in the low-dim case.

3 Wine dataset analysis

In order to test and better understand the main differences between PCA and ROBPCA algorithms and how each approach behave in the presence of outliers and in their detection, we used a dataset with information about wine's parameters and their respective quality and applied these algorithms to the data. An analysis about the outliers of this dataset has also been done. For the computation of the PCA it was used the function *PCAClassic* of the package *rrcov* [5], and for the ROBPCA it was used the function *robpca* of the package *rospca*[3].

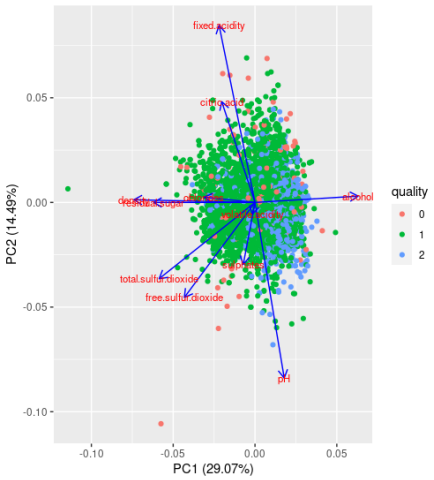

Figure 6: White wine dataset and original features projection on the subspace generated by the first 2 PC's.

3.1 The dataset

The data were collected from May 2004 to February 2007 from the demarcated region of *vinho verde* (green wine) in Minho, in the north-west of Portugal [6], and comprises 11 physicochemical variables of numeric type, shown in Table 3, and one output/target variable, of categorical type. The target variable is the quality grade, based on sensory data and determined by the median of at least three blind evaluations made by wine experts. Each expert graded the wine quality between 0 (very bad) and 10 (very excellent). The white wine dataset has a total of 4898 samples. Instead of using the seven classes we decided to create three new classes by aggregation of quality grades namely: class 0 (= grades 3 and 4) denoting "bad quality"; class 1 (= grades 5 and 6) denoting "medium quality"; and class 2 (= grades 7, 8 and 9) denoting "good quality".

By analysing the percentage of Cumulative Variance (figure 7), 6 PC's are enough to explain at least 80% of the total variance, both on PCA and on the ROBPCA cases, and so it was decided to perform the outlier analysis with the space spanned by these first 6 PC's. Nonetheless, it is worth mentioning that the PC's created by ROBPCA were capable of explaining a slightly more variability than the respective PC's produced by the classical PCA.

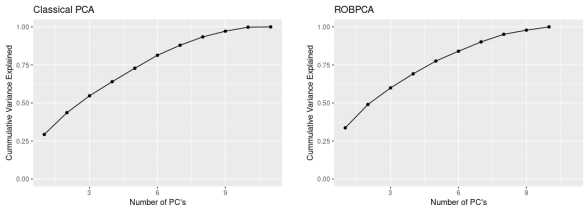

Figure 7: Percentage of Variance Explained by each PC for the classical PCA and ROBPCA.

3.2 Results and discussion

A first PCA analysis was done to the original dataset, and projection of the features on the subspace generate by the first 2 PC's can be seen in figure 6, and the most relevant features of each PC can be seen in table **??**. By examining figure 6 we can have a first a first perception of the existence of some observations that lie far away from the regular observations in the subspace generated by the first two PC's. Notice that by examining this projection we can already have a some intuition of which features might be more determinant in what regards the outliers scores.

For the diagnostics plot is was used a false alarm rate of 0.999 (which corresponds to the alpha parameter used in the function robpca), since there is a wide variability of wines, and given that wine quality is a highly subjective parameter; that being said, it was of our most interest to not classify too many wines as outliers. The Diagnostic Plot for both PCA and ROBPCA, using 2, 6 and 9 PC's can be seen in figures 8, 9 and 10, respectively.

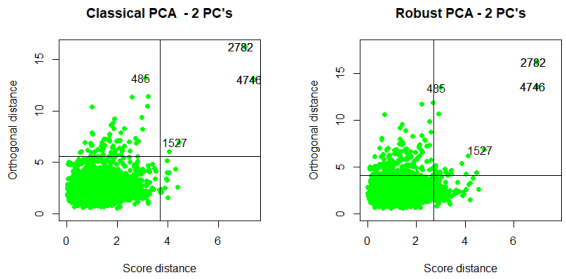

Figure 8: Diagnostic Plot for PCA and ROBPCA using 2 PC's.

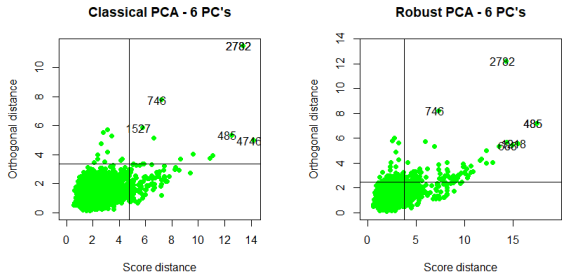

Figure 9: Diagnostic Plot for PCA and ROBPCA using 6 PC's.

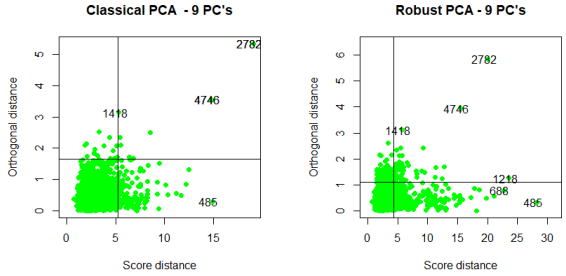

Figure 10: Diagnostic Plot for PCA and ROBPCA using 9 PC's.

From the plot's inspection, we can see that the use of a higher number of PC's decreased both the cutoff values of the OD and the SD, and so it increased the number of observations classified as bad leverage points, as well as it increased the number of observations classified as orthogonal outliers an good leverage points. By increasing the number of PC's, the PC subspace generated increases it's dimension, and so points that didn't lie in the smaller subspace may now lie in this subspace; points that were in the smaller PCA subspace belong to the larger PC thus explaining the lower cutoff value for the OD and the SD, both for the PCA and the ROBPCA.

Comparing the results from ROBPCA and PCA, we can see that ROBPCA has consistently assigned more observations as outliers than the PCA, for every number of PC's that was tried. In every case, many of the observations assigned by ROBPCA as bad leverage points were classified as good leverage point when applying the PCA. This suggests how much is the PCA subspace is attracted by the points with grater OD's, and why ROBPCA is more robust to the presence of these types of observations than PCA.

For the case where 6 PC's were used, the PCA classified as outliers a total of 13 observations against 98 observations classified as outliers by the ROBPCA. Information regarding the statistics about the Wine dataset, and the outliers produced by PCA and ROBPCA can be seen in tables 5, 6 and 7, accordingly. Every point identified as an outlier by PCA was also identified as an outlier by ROBPCA. By taking a closer look to the tables, we can see that wines classified as bad (quality $= 0$) were over represented in the outliers group, wines classified as good (quality $= 2$) were underrepresented. This may indicate some correlation between the quality of the wine and the wine being an outlier. Both on the PCA a ROBPCA outliers, total and free sulfur dioxide has an higher mean than in the original data set. This may be the reason why the outliers have also a higher mean value of fixed acidity, since the in order to regulate the pH of the may be the need to add sulphates to the wine, so that further fermentation can be prevented. It is also interesting to notice that the wines identified as outliers by PCA have a higher value of residual sugar, but the mean residual sugar of the outliers detected by ROBPCA is close to the one of the original dataset. Both residual sugar and total sulfur dioxide are main features with negative coefficients in the 1st PC, which alone explains approximately 30% of the total variability; that being said, it is natural that the observations identified as outliers higher distinctive values for these two parameters.

4 Conclusions

The results from the simulation replication and from the wine dataset analysis indicate that ROBPCA distinguishes mainly from PCA by being a robust method against observations with a great orthogonal distance.

As for the simulation we observed the subspace generated by the first k eigenvectors from CPCA deviates substantially from the underlying k-dim subspace of the regular data, which is not the case of the subspace generated by the first k eigenvectors from ROBPCA. Thus, ROBPCA shows experimental evidence of its robustness relative to the presence of outliers in the contamination model. Concerning the mean proportion of explained variability we observed stable results for ROBPCA and more deviating percentages for CPCA, across the various values of the parameters considered. With respect to the MSE's ratios we observed the presence of considerably higher MSE's ratios in at least on of the eigenvalues and this explains the *maxsub* values obtained. As we obtained some unexpected values in the simulation, except in *maxsub*, as well as taking in comparison the results in [1], we may infer this could be due to the fact that as we are using pre-developed functions from R packages for CPCA and ROBPCA and not the case of the implemented algorithms in [1], results may show differences from what is expected. If implementing from scratch a more thorough and careful production of results and their analysis could be better performed.

As for the White Wine dataset, from the performed outlier analysis, residual sugar and total free sulfur dioxide seem be important features in what regards the classification of an observation as an outlier. A closer inspection of the outliers is needed in order to identify which other features may play a larger role in identifying outlier wines. The methodology used in this work also suggests a one possible way to better comprehend the most decisive factors when it comes to evaluate a wine quality, given that bad wines were over represented in the outliers group.

5 Appendix

		CPCA	ROBPCA
$\mathbf{f}_1 = 6$	**n = 100, p = 4**		
	$\epsilon = 0\%$	94.76%	95.60%
	$\epsilon = 10\%$	77.97%	78.47%
	$\epsilon = 20\%$	69.93%	65.10%
	n = 50, p = 100		
	$\epsilon = 0\%$	92.91%	99.74%
	$\epsilon = 10\%$	76.27%	81.95%
	$\epsilon = 20\%$	65.03%	67.74%
$\mathbf{f}_1 = 10$	**n = 100, p = 4**		
	$\epsilon = 0\%$		
	$\epsilon = 10\%$	79.90%	78.35%
	$\epsilon = 20\%$	87.05%	64.87%
	n = 50, p = 100		
	$\epsilon = 0\%$		
	$\epsilon = 10\%$	76.84%	81.98%
	$\epsilon = 20\%$	70.45%	67.74%
$\mathbf{f}_1 = 14$	**n = 100, p = 4**		
	$\epsilon = 0\%$		
	$\epsilon = 10\%$	85.75%	78.43%
	$\epsilon = 20\%$	112.64%	65.03%
	n = 50, p = 100		
	$\epsilon = 0\%$		
	$\epsilon = 10\%$	78.87%	81.78%
	$\epsilon = 20\%$	78.84%	67.92%
$\mathbf{f}_1 = 18$	**n = 100, p = 4**		
	$\epsilon = 0\%$		
	$\epsilon = 10\%$	94.30%	78.45%
	$\epsilon = 20\%$	147.31%	64.95%
	n = 50, p = 100		
	$\epsilon = 0\%$		
	$\epsilon = 10\%$	81.63%	81.86%
	$\epsilon = 20\%$	89.92%	67.78%

Table 2: Mean proportion of explained variability results for the simulation data, for each value of f_1, each contamination factor ϵ, for low-dim and high-dim cases, *case 1* and *case 2*, respectively, and for $f_2 = 1$. The blank spaces are intentional, as for no contamination there is no dependency on the parameters f_1 and f_2.

	feature		feature		feature
1	Fixed acidity (g/dm^3)	**2**	Volatile acidity (g/dm^3)	**3**	Citric acid (g/dm^3)
4	Residual sugar (g/dm^3)	**5**	Chlorides (g/dm^3)	**6**	Free $SO_2(mg/dm^3)$
7	Total $SO_2(mg/dm^3)$	**8**	Density (g/dm^3)	**9**	pH
10	Sulphates (g/dm^3)	**11**	Alcohol (vol %)		

Table 3: Physicochemical parameters of wine with respective numbering of features.

PC1	(+) density , (+) alcohol, (-) residual sugar, (-) total sulfur dioxide
PC2	(+) fixed acidiy , (-) pH
PC3	(+) volatile acidity, (-) citric acid, (-) sulphates
PC4	(+) chlorides, (-) free sulfur dioxide, (+) sulphates
PC5	(-) sulphates, (+) chlorides, (-) volatile acidity, (-) fixed acidity
PC6	(+) free sulfur dioxide, (+) total sulfur dioxide, (+) volatile acidity, (-) density

Table 4: Main coefficients on each Principal Component. Every feature is represented in at least one of the PC's.

Table 5: Summary statistics for the White Wine dataset

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
fixed.acidity	4898	6.855	0.844	3.8	6.3	7.3	14.2
volatile.acidity	4898	0.278	0.101	0.08	0.21	0.32	1.1
citric.acid	4898	0.334	0.121	0	0.27	0.39	1.66
residual.sugar	4898	6.391	5.072	0.6	1.7	9.9	65.8
chlorides	4898	0.046	0.022	0.009	0.036	0.05	0.346
free.sulfur.dioxide	4898	35.308	17.007	2	23	46	289
total.sulfur.dioxide	4898	138.361	42.498	9	108	167	440
density	4898	0.994	0.003	0.987	0.992	0.996	1.039
pH	4898	3.188	0.151	2.72	3.09	3.28	3.82
sulphates	4898	0.49	0.114	0.22	0.41	0.55	1.08
alcohol	4898	10.514	1.231	8	9.5	11.4	14.2
quality	4898						
... 0	183	3.7%					
... 1	3655	74.6%					
... 2	1060	21.6%					

Table 6: Summary Statistics for the outliers detected by PCA

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
fixed.acidity	13	7.585	2.08	6.1	6.6	7.6	14.2
volatile.acidity	13	0.377	0.234	0.19	0.2	0.37	0.965
citric.acid	13	0.548	0.466	0.14	0.25	0.6	1.66
residual.sugar	13	9.854	17.256	1.1	2.1	8.3	65.8
chlorides	13	0.167	0.117	0.022	0.047	0.271	0.346
free.sulfur.dioxide	13	54.385	72.524	8	24	45	289
total.sulfur.dioxide	13	194.077	87.363	113	142	200	440
density	13	0.998	0.013	0.99	0.993	0.996	1.039
pH	13	3.135	0.168	2.93	3.03	3.26	3.44
sulphates	13	0.527	0.113	0.31	0.45	0.63	0.69
alcohol	13	10.3	1.352	8.7	9.4	11.1	13.1
quality	13						
... 0	2	15.4%					
... 1	11	84.6%					

Table 7: Summary Statistics for the outliers detected by ROBPCA

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
fixed.acidity	98	7.128	1.395	4.5	6.3	7.6	14.2
volatile.acidity	98	0.386	0.21	0.17	0.24	0.459	1.1
citric.acid	98	0.483	0.254	0.14	0.3	0.653	1.66
residual.sugar	98	6.647	8.508	0.8	1.562	8.45	65.8
chlorides	98	0.123	0.08	0.022	0.047	0.184	0.346
free.sulfur.dioxide	98	47.372	38.977	5	24.25	60	289
total.sulfur.dioxide	98	164.189	66.335	54	120	192.25	440
density	98	0.995	0.006	0.987	0.993	0.997	1.039
pH	98	3.125	0.152	2.87	3.01	3.218	3.55
sulphates	98	0.536	0.169	0.31	0.422	0.628	1.08
alcohol	98	10.011	1.114	8	9.3	10.5	14
quality	98						
... 0	20	20.4%					
... 1	72	73.5%					
... 2	6	6.1%					

References

[1] Rousseeuw P. n Van Den Branden K. Hubert, M. Robpca: A new approach to robust principal component analysis. *Technometrics*, 47(1):64–79, 2005.

[2] Support functions and datasets for venables and ripley's mass, 2022. URL `https://cran.r-project.org/web/packages/MASS/MASS.pdf`.

[3] Robust sparse pca using the rospca algorithm, 2018. URL `https://cran.r-project.org/web/packages/rospca/rospca.pdf`.

[4] prcomp: Principal components analysis, 2022. URL `https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/prcomp`.

[5] Valentin Todorov. Scalable robust estimators with high breakdown point [r package rrcov version 1.7-0]. 2022.

[6] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4):547–553, 2009.