A Bias Trick for Centered Robust Principal Component Analysis (Student Abstract)

Baokun He, Guihong Wan, Haim Schweitzer
{Baokun.He, Guihong.Wan, Haim}@utdallas.edu
The University of Texas at Dallas
800 W. Campbell Road, Richardson, Texas

Abstract

Outlier based Robust Principal Component Analysis (RPCA) requires centering of the non-outliers. We show a "bias trick" that automatically centers these non-outliers. Using this bias trick we obtain the first RPCA algorithm that is optimal with respect to centering.

1 Introduction

Principal Component Analysis (PCA) is arguably the most widely used dimensionality reduction technique. It is known that the PCA model is heavily influenced by data outliers. The detection and removal of such outliers is a key component of robust variants of PCA.

There are two main variants of standard PCA: centered and uncentered. The only difference between them is that in centered PCA there is a preliminary step where the data is being centered. From a computational point of view there is little difference between these two variants. For this reason most recently published fast algorithms for computing PCA ignore the centering of the data. The situation is very different for algorithms that attempt to compute Robust PCA (RPCA) by identifying some points as outliers to be removed. The problem is that the centering should be applied only to the non-outliers, but they are initially unknown.

Some previously proposed RPCA algorithms perform initial centering of the data but do not update the center based on the outliers. These include (Zhang et al. 2015; Xu et al. 2010; Shah et al. 2017). Other algorithms such as (Xu,et al. 2013; Rahmani and Atia 2017) do not explicitly center the data. The first assumes a probability distribution of the mean, and the second does not consider the magnitude but only angles which makes centering unnecessary. Other algorithms such as (Hubert and Engelen 2004) handle the centering as part of the algorithm, but not optimally. This review of the current state of the art suggests that optimal centering in RPCA is not fully solved.

We propose a general method (a bias trick) that can be used to convert *any* robust algorithm that does not perform centering into an algorithm that performs centering optimally. In fact, the bias trick can be used to convert any algorithm that computes uncentered PCA into an algorithm that computes a centered PCA.

Using the bias trick with the algorithm of (Shah et al. 2017) that computes optimal uncentered RPCA gives the first optimal centered RPCA algorithm. We implemented this algorithm and describe some experimental results, showing improved performance over all competitors.

2 The Bias Trick

Let $\mathsf{PCA}()$ be an uncentered PCA algorithm. It gets as input the matrix X of size $m\times n$ and the number k of desired principal vectors. It returns the principal vectors as $v_1, ..., v_k$ and the corresponding eigenvalues. To apply the bias trick and obtain the centered PCA we do the following:

1. Select a large value b. (See Section 3.)
2. Add b as an additional coordinate to each column of X, creating a new matrix X_b of size $(m+1)\times n$.
3. Run $\mathsf{PCA}()$ on X_b to compute $k+1$ eigenvectors and eigenvalues. Each eigenvector is of size $(m+1)$.
4. Let $\lambda_1^b \ldots, \lambda_{k+1}^b$ be the eigenvalues computed in Step 3. Then the k eigenvalues of the centered PCA are approximately $\lambda_2^b \ldots \lambda_{k+1}^b$.
5. Let $u_1^b \ldots, u_{k+1}^b$ be the eigenvectors computed in Step 3. Let v_j be the the jth eigenvector of the centered PCA. It is given approximately by the top m values of u_{j+1}^b.

Clearly, the bias trick is not an improvement over the standard centered PCA algorithm. It is more costly and less accurate. But, it has the advantage that it also works for centered RPCA where it does not require advanced knowledge of the outliers. Applying the bias trick for computing centered RPCA can be achieved by using $\mathsf{RPCA}()$ instead of $\mathsf{PCA}()$, where $\mathsf{RPCA}()$ is any uncentered RPCA algorithm.

3 Correctness of the Bias Trick

The following theorem is proved in (He et al. 2019):

Theorem: Let X be the data matrix and let μ be the column mean. For any desired accuracy of computing the centered PCA there exists $0<\epsilon<1$ such that setting $b \geq \frac{\sqrt{1-\epsilon^2}}{\epsilon}\|\mu\|$

in the procedure outlined in Section 2 gives the desired approximation.

4 Experiments

Effect of The Bias Value. In the first experiment we demonstrate that centered PCA implemented with the bias trick returns accurate eigenvalues. The error on "iris" (from UC Irvine) for various ϵ values is shown in Figure 1. Observe that for moderate values of ϵ the error is almost 0. Similar results were obtained with other datasets, suggesting that $\epsilon \approx 0.2$ (or equivalently $b \approx 5\|\mu\|$) may give sufficient accuracy.

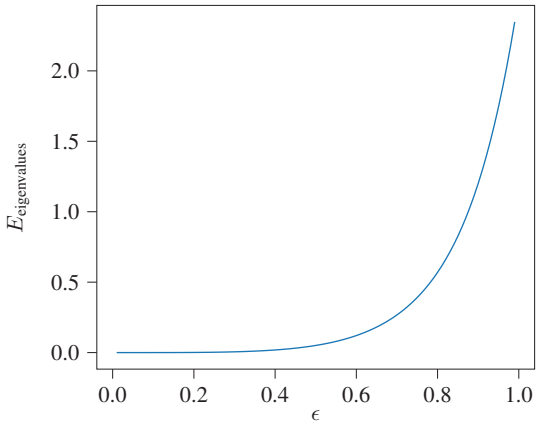

Figure 1: Eigenvalue error with the bias trick on "iris" as a function of ϵ. The error is almost 0 for ϵ below 0.2.

Optimal Centered Robust PCA. We describe the results of using the algorithm of (Shah et al. 2017) with the bias trick. The original algorithm computes optimal uncentered RPCA. With the bias trick the algorithm computes optimal centered RPCA. To the best of our knowledge this is the first centered algorithm with guaranteed optimality. We refer to this algorithm as COPT.

Tables 1 and 2 show errors for two algorithms using code provided by the authors. The values are the average reconstruction error of the n non-outlier points: $E_{\text{rpca}} = \frac{1}{n}\sum_i ||(x_i - \mu) - V_{\text{rpca}} V_{\text{rpca}}^T (x_i - \mu)||^2$. Our COPT error is always smaller.

Table 1: Comparison to Outlier-Pursuit (Xu et al. 2010)

dataset	$k:r$	ϵ	COPT	Outlier-Pursuit
smoking	2:1	0.2	703.55	1159.70
wdbc	20:2	0.2	241.46	304.24
wine	5:2	0.2	14.72	15.50

Figures 2 and 3 compare the results of our COPT algorithm to the results of the Outlier-Pursuit algorithm. Five outliers were selected based on the first two principal vectors. The left panel in both figures shows the location of all the points in the plane defined by these vectors. The right panel is the location of all the points on the plane defined by the first and third principal vectors. It clearly shows the outliers at the margins of the distribution. By contrast, in Figure 3 the outliers are not the ones farthest away. This is a further evidence that our COPT compares favorably to the Outlier-Pursuit algorithm.

Table 2: Comparison to CoP (Rahmani and Atia 2017)

dataset	$k:r$	ϵ	COPT	CoP
smoking	1:1	0.2	1343.00	1343.00
wdbc	17:2	0.2	252.14	498.75
wine	5:2	0.2	14.72	15.50

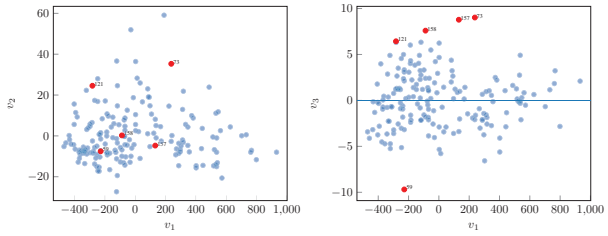

Figure 2: Outliers selected by our COPT algorithm on "wine". (Red points are the outliers, k=5, r=2, E_{rpca}=14.72).

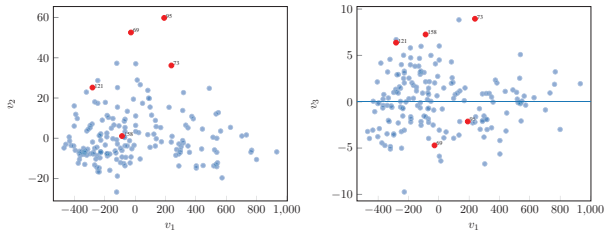

Figure 3: Outliers selected by the Outlier-Pursuit algorithm on the centered "wine". (Red points are the outliers, k=5, r=2, E_{rpca}=15.5).

References

Hubert, M., and Engelen, S. 2004. Robust PCA and classification in biosciences. *Bioinformatics* 20(11):1728–1736.

Rahmani and Atia 2017. Coherence pursuit: Fast, simple, and robust principal component analysis. *IEEE TSP* 65(23):6260–6275.

Shah, S.; He, B.; Maung, C.; and Schweitzer, H. 2018. Computing robust principal components by A* search. *IJAIT* 27(7).

Xu, H.; Caramanis, C.; and Mannor, S. 2013. Outlier-robust pca: The high-dimensional case. *IEEE Transactions on Information Theory* 59(1):546–572.

Xu, H.; Caramanis, C.; and Sanghavi, S. 2010. Robust pca via outlier pursuit. In NIPS'10, 2496–2504.

Zhang, H.; Lin, Z.; Zhang, C.; and Chang, E. Y. 2015. Exact recoverability of robust pca via outlier pursuit with tight recovery bounds. In AAAI'15, 3143–3149.

He, B.; Wan, G.; and Schweitzer, H. 2019. A Bias Trick for Centered Robust Principal Component Analysis. arXiv:1911.08024.