A New Robust Subspace Recovery Algorithm

Guihong Wan, Haim Schweitzer
{Guihong.Wan, HSchweitzer}@utdallas.edu
The University of Texas at Dallas
800 W. Campbell Road, Richardson, Texas, 75083

Abstract

A common task in data analysis is to compute an approximate embedding of the data in a low dimensional subspace. This is used, for example, for dimensionality reduction. Robust Subspace Recovery computes the embedding by ignoring a fraction of the data considered as outliers. Its performance can be evaluated by how accurate the inliers (non-outliers) are represented. We propose a new algorithm that outperforms the current state of the art when the data is dominated by outliers. The main idea is to rank each point by evaluating the change in the global PCA error when that point is considered an outlier. We show that this lookahead procedure can be implemented efficiently by centered rank-one modification.

1 Introduction

Principal Component Analysis (PCA) is a common technique for embedding data in a low dimensional subspace. See, e.g., (Jolliffe 2002). Unfortunately, it is known to be sensitive to outliers. Robust algorithms identify the outliers and produce embedding only for the inliers. For recent surveys that discuss many variants of these robust algorithms see (Lerman and Maunu 2018; Vaswani and Narayanamurthy 2018). The second reviews techniques that consider outliers as partially corrupt coordinates, while the first reviews techniques that consider each point as either an outlier or an inlier. We follow the same interpretation of outliers as in the first reference.

Typically the term "outliers" refers to a small portion of the data that does not follow the same pattern as most of the data. We consider situations where a significant portion of the data is irrelevant. The algorithmic challenge of dealing with large fractions of outliers is quite different from handling a small fraction of outliers. The first issue is running time, which may significantly increase with the number of outliers. The second issue is accuracy. If there are many outliers they may mask the statistical properties of the inliers. Indeed, evaluating the performance of previously proposed robust subspace recovery algorithms we observed significant deterioration in accuracy when increasing the outlier fraction.

Our approach

A fundamental part in our approach is to assign a value to each point, indicating how likely it is to be an outlier. We compute this as the error of the entire PCA model "if" that point becomes an outlier. Our technical contribution is showing that these lookahead errors can be computed efficiently by a rank-one modification of centered matrices. Our main contributions are: 1. An algorithm for fast eigenvalue updates of centered matrices. 2. A Robust Subspace Recovery algorithm based on 1.

2 Top view of the algorithm

Let $X=(x_1, \ldots, x_n)$ be the data matrix of n data points and each point is of dimension m. Let r be the desired number of principal components. Let O_1, O_2 be two outlier subsets of same size: $|O_1|=|O_2|$. Suppose we have access to a function f such that: if $f(O_1, X, r)>f(O_2, X, r)$, then O_1 "appears to be" a better outlier set than O_2. Using f we propose an iterative algorithm for computing k outliers. Given an outlier set O, a single iteration that adds c outliers to O is given below:

Input: Data X, a set O of j outliers,
user defined parameters α, k, r.
1. For each i=1...$n-j$, create the child O_i by adding the inlier x_i to O. Compute $f_i=f(O_i, X, r)$.
2. Compute c from α by: $c = \alpha(k-j-1)+1$.
Output: union O with the c children of largest f_i.
As a formula: $O = \text{Update}(O, \alpha, k, X, r)$.

We also use a "k-means" style algorithm as a subroutine:

Input: Data X, a set O of j outliers,
user defined parameter r.
Initialization: Set current error value to infinity.
Repeat:
1. Set old error value to the value of current error.
2. Compute V, μ from rank-r PCA of the inliers.
3. $e_i = ||(x_i-\mu) - VV^T(x_i-\mu)||^2$ for all x_i.
4. Replace O by the j columns with the largest e_i.
5. Set current error value = $\sum_{x_i \notin O} e_i$.
Until: current error value = old error value.
Output: the new set O of j outliers.
As a formula: $O = \text{KM}(O, X, r)$.

Iterating these algorithms gives the entire algorithm:

Input: $1 \le k \le n, \quad 0 \le \alpha \le 1, \quad X, \quad r.$
Output: A subset O of k outliers.
Initialization: $O = \emptyset$ (the empty set).
Iterate: while $|O|<k$ do:
O=Update(O, α, k, X, r), O=KM(O, X, r).

The value of α affects both the accuracy and the running time of the algorithm. Increasing α would in general result in a reduction in accuracy and a faster running time. Correctness proofs are given in the full paper.

3 The lookahead error

Given O and r, the PCA error is defined as the summation of reconstruction errors for all inliers:

$$e_i = \|(x_i - \mu) - VV^T(x_i - \mu)\|^2, \quad E(X, O) = \sum_{i \notin O} e_i \quad (1)$$

where $\mu = \frac{1}{n-|O|} \sum_{i \notin O} x_i$. It is known (e.g., (Jolliffe 2002)) that V has as its columns the r dominant eigenvectors of the covariance matrix C of inliers. Our results rely on a relationship between the PCA error and the eigenvalues of C.

Theorem 1: (The proof is given in the full paper.)

$$E(X, O) = \sum_{i=r+1}^{n} \lambda_i(C) = \text{trace}\{C\} - \sum_{j=1}^{r} \lambda_j(C)$$

The lookahead error of a point x_i is the model error E as defined in (1) of O_i, obtained by adding x_i to the current selected outlier set O: $f_i = f(O_i, X, r) = E(X, O_i)$.

4 Rank-one modification

A direct evaluation of the lookahead error in the "Update" steps requires that a PCA algorithm is applied to each inlier. This is impractical. We show how to calculate lookahead errors efficiently. Let p be the number of inliers at the parent level. Let Z be the $m \times p$ matrix of the inliers and μ be the column mean of Z. Then the scaled covariance matrix is: $C = Z^c(Z^c)^T$, where $Z^c = Z - \mu \mathbf{1}^T$. Suppose a child is constructed by removing x_i from Z. Let Z_i be the $m \times (p-1)$ matrix of the remaining inliers and μ_i be the mean of Z_i. The following Theorem shows that $C_i = Z_i^c(Z_i^c)^T$ can be obtained by a rank-one modification of C. To the best of our knowledge this was not previously observed.

Theorem 2: (The proof is given in the full paper.)
Define: $y_i = x_i - \mu$, and $\beta = \frac{p}{p-1}$. Then: $C_i = C - \beta y_i y_i^T$.

Corollary: $\text{trace}\{C_i\} = \text{trace}\{C\} - \beta \|y_i\|^2$.

The rank-one modification can be used to obtain fast calculations of eigenvalues. See, e.g., (Bunch, Nielsen, and Sorensen 1978).

5 Experimental results

We follow closely the experimental methodology used in the recent survey of the field (Lerman and Maunu 2018). It uses two artificial datasets (Haystack and Blurryface) to compare algorithms. We experimented on these datasets with the same error criteria. We refer to our lookahead algorithm

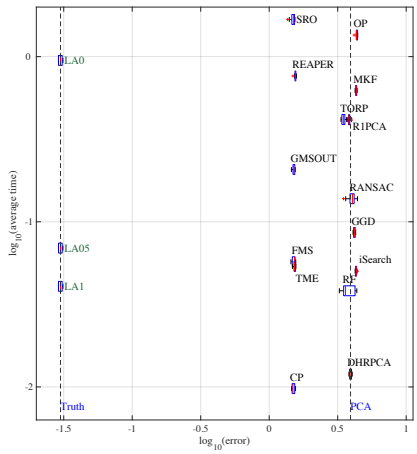

Figure 1: Box plots for the Haystack. Outlier fraction: 0.6. Outlier mean: 1. Time was averaged over 10 runs.

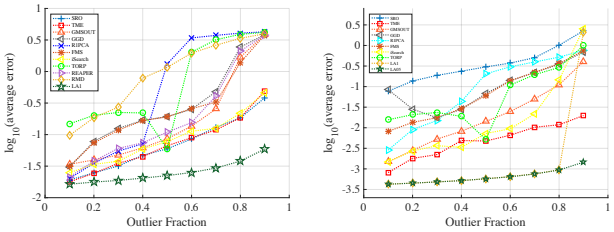

Figure 2: Error versus outlier fraction. Outlier mean: 0. Left: Haystack. Right: Blurryface.

with the initials LA followed by the α value. Thus, LA05 means the algorithm with α=0.5. The comparison was with other algorithms that are detailed in the full paper. We added "Truth" as the ground truth result, and "PCA" for the result of (non-robust) standard PCA.

The results show a huge decline in the accuracy of current state-of-the-art algorithms on the nonzero mean data shown in Figure 1. The vertical axis corresponds to speed, and the horizontal axis to accuracy. We observe that the lookahead variants are very accurate, but none of the other algorithms are. In fact, many have similar accuracy to the (non robust) standard PCA.

Plots of the accuracy as a function of the outlier fraction are shown in Figure 2. For Blurryface, LA05 and LA1 are distinctly better than the other algorithms until 80% after that the accuracy of LA1 deteriorates.

References

Bunch, J. R.; Nielsen, C. P.; and Sorensen, D. C. 1978. Rank-One Modifications. *Numer. Math.* 31: 31–48.

Jolliffe, I. T. 2002. *Principal Component Analysis*. New York: Springer-Verlag, second edition.

Lerman, G.; and Maunu, T. 2018. An Overview of Robust Subspace Recovery. *Proceedings of the IEEE* 106(8).

Vaswani, N.; and Narayanamurthy, P. 2018. Static and Dynamic Robust PCA. *Proc. IEEE* 106(8): 1359–1379.