Grassmann Averages for Scalable Robust PCA
Søren Hauberg, Aasa Feragen and Michael J. Black
DTU Compute
Technical University of Denmark
Dept. of Computer Science
University of Copenhagen
MPI for Intelligent Systems
Tübingen, Germany

1) Motivation: Scalable robust statistics
We are collecting increasing amounts of data in a largely automated way. The large-scale harvesting increases the odds of collecting unwanted data. In other words, we should expect increasing amounts of outliers when automating data collection.

We, thus, need algorithms which can both cope with large amounts of data and deal with many outliers.

We investigate large-scale robust PCA.

2) Idea #1: Match input and output types
The simplest and most well-understood statistical estimators are those where the "type" of the input and output match. Example: the standard average (input: points; output: a point).

The input is a set of points

The desired output is a T-dimensional subspace

The input is the average subspace

The desired output is an estimator

Corresponding types will simplify both math and algorithms. Many estimators are now easily phrased.

We study the average subspace.

3) Representing subspaces
The most obvious representation of 1D subspaces is a unit vector spanning the space. Changing the sign of a unit vector does not change the space being spanned, so our representation has an unknown sign.

This is a simple instance of the general Grassmann manifold, which represents higher dimensional subspaces.

4) Grassmann average algorithm
A first result for deriving an algorithm is

\[ a = \arg \min \sum \frac{d}{d_{ij}}, \quad \text{where} \quad d_{ij} = (u - v)^T u \quad \text{over all pairs} \quad (u, v). \]

With an simple Euclidean distance measure

\[ d_{ij} = (u - v)^T u \]

the following algorithm naturally appears:

Algorithm: Grassmann average (GA)

\[ a = \arg \min \sum \frac{d_{ij}}{d_{ij}}, \quad \text{where} \quad d_{ij} = (u - v)^T u. \]

The algorithm optimises this robust energy:

\[ a = \arg \min \sum \frac{d_{ij}}{d_{ij}}, \quad \text{also known as} \quad \text{L1-PCA}\text{, } \text{Krum, TRIM, 2008}. \]

This is more robust than the standard PCA energy:

\[ \text{PCA: } \arg \min \sum \frac{d_{ij}}{d_{ij}}, \quad \text{where} \quad d_{ij} = (u - v)^T u. \]

5) Idea #2: Averages are easy to make robust
One of the most well-understood robust statistics is the robust average.

As we are merely computing a subspace average, we can easily phrase a robust Grassmann average:

Algorithm: Robust Grassmann Average (RGA)

\[ a = \arg \min \sum d_{ij}, \quad \text{where} \quad d_{ij} = (u - v)^T u. \]

In computer vision we are mostly concerned with pixel-level outliers, so we suggest a simple pixel-level trimmed average. This can be computed with the same complexity as a standard average.

6) Results
The right figure investigates the statistical efficiency of the proposed estimators. We note the standard robustness-efficiency tradeoff.

Below we consider a film restoration task where noisy frames are denoted by projecting to a robust (GA) subspace.

The estimated noise is transferred to new films to provide quantitative results.

We further model backgrounds with changing light using a robust subspace.

Finally, we show scalability by computing the leading 3D robust components of the entire Star Wars IV movie (a task beyond other methods).

More Information
All out web page have source code (Matlab and C+++) along with further results, the paper and its supplementary material.
http://ps.is.tue.mpg.de/project/Robust_PCA