

Comparison of robust PCA techniques in image and video processing

Joel Loo, Wang Xueqiang

1 Introduction

Principal components analysis (PCA) and other dimensionality reduction techniques have gained wide traction in the scientific community because of their utility in extracting salient data from large and complex datasets. However, PCA implicitly assumes that the noise in data is small and variance from the mean is bounded (such as in Gaussian noise), causing it to perform poorly on grossly corrupted data, or data containing outliers of significant magnitude.

While there have been many attempts to robustify PCA, many of these methods are heuristics without any optimality guarantees. However, recent advances in low-rank matrix recovery and completion by Candès et al [1], have led to the formulation of robust PCA (RPCA). In RPCA, the problem is reformulated as the recovery of the matrices L, S such that $M = L + S$, where M is the original matrix, L is a low-rank matrix and S is a sparse matrix. Candès et al made the surprising observation that it is possible to perform this decomposition through a tractable convex optimization, if we assume that the low-rank matrix is not simultaneously sparse [1]. In particular, this can be achieved using the Principal Component Pursuit (PCP) estimate which solves the problem

$$\operatorname{argmin}_{L, S} \|L\|_* + \lambda \|S\|_1 \quad \text{subject to} \quad L + S = M$$

Since the sparse matrix can have elements of arbitrarily large magnitude, robust PCA thus improves over the PCA technique by allowing the extraction of low-dimensional structure from grossly corrupted data with unbounded outliers. This technique has many important applications in general, and in particular is of great utility in computer vision, where it is used in tackling problems such as background recovery/subtraction (which can be used in video surveillance software), removal of non-Lambertian distortions/specularities/shadows from images (akin to performing inpainting) and even recovery of 3D geometry from low-rank textures in 2D images.

In order to make RPCA practicable on real-world vision systems, a number of reformulations of the problem and improvements have been proposed. Some of these improvements (e.g. Fast PCP [3]) provide significant speed improvements over the method proposed by Candès et al, while others reformulate RPCA as an online algorithm (e.g. online RPCA via stochastic optimization [2], iFrALM).

2 Problem Statement

In this project, we aim to survey several RPCA algorithms - in particular we will implement these algorithms and compare their performance, in terms of both speed and ability to separate low-rank structure from sparse noise. We are primarily interested in applying these algorithms to, and testing them on vision applications. Two applications that we are particularly interested in are the removal of specularities/non-Lambertian distortions from images, as well as background recovery/subtraction in videos. Background

recovery/subtraction in video streams provides an interesting case study for RPCA as well, since it is desirable to have both algorithms that can separate backgrounds in post-processing as well as on the fly. To this end, we will use this particular application to benchmark the performance of both RPCA algorithms that offer significant speed-ups, as well as RPCA algorithms formulated to be online in nature. We will perform the implementations and benchmarking described above in Python, with code available on the private Github repository https://github.com/joelloo/18660_project

3 Methods

In this section, we will discuss the formulation of the optimization problems for the various algorithms and our approach to their implementation. We first approach RPCA algorithms that only enable offline computation in a video stream background subtraction setting (i.e. require all the frames of the video for the optimization), before considering online RPCA algorithms that enable on-the-fly updating of the low-rank/sparse decomposition as new frames come in from the video stream.

3.1 Offline RPCA Algorithms

The original PCP formulation of the RPCA algorithm has several immediately apparent solutions. We can consider the relaxation $\operatorname{argmin}_{L,S} \|L\|_* + \|S\|_1 + \frac{1}{2} \|M - L - S\|_F^2$. Since $\|\cdot\|_*$ and $\|\cdot\|_1$ have well-known proximal operators in the singular value thresholding operator and the soft thresholding operator, while $\|\cdot\|_F$ is differentiable, it is clear that we can solve this formulation via proximal gradient descent. Indeed, Lin et al. [5] provide an accelerated proximal gradient descent (APG) based algorithm that solves this formulation by alternately minimizing L (with singular value thresholding) then E (with soft thresholding) at each iteration.

Another possible means of approaching the PCP formulation is via the augmented Lagrangian multiplier method (ALM), as suggested by Candes et al. in [1] and elaborated on by Lin et al. in [8]. We can write the Lagrangian function of the PCP formulation as $\mathcal{L}(L, S, Y, \mu) = \|L\|_* + \lambda \|S\|_1 + \langle Y, M - L - S \rangle + \frac{\mu}{2} \|M - L - S\|_F^2$. The augmented Lagrangian multiplier method then involves solving the Lagrangian function as shown in the

Algorithm 1 Augmented Lagrangian multiplier method

```

Initialize  $Y_0 = \operatorname{sgn}(D)/J(D)$ ;  $\mu_0 > 0$ ;  $\rho > 1$ ;  $k = 0$ 
while not converged do
   $(L_{k+1}^*, S_{k+1}^*) = \operatorname{argmin}_{L,S} \mathcal{L}(L, S, Y_k^*, \mu_k)$ 
   $Y_{k+1}^* = Y_k^* + \mu_k (M - L_{k+1}^* - S_{k+1}^*)$ 
  Update  $\mu_k$  to  $\mu_{k+1}$ 
end while
return  $L_k, S_k$ 

```

algorithm above. gi

3.2 Online RPCA Algorithms

4 Preliminary results and comparisons

5 Next steps