ROBUST PRINCIPAL COMPONENT ANALYSIS AND ITS APPLICATIONS

Gaurav Kumar (14240), Bhargav Ganguly (14807177), Gorantala Sai Teja (14250)

Indian Institute of Technology, Kanpur

ABSTRACT

PCA algorithm is used to recover low rank from the data matrix. Robust PCA is PCA with slight modifications which works well even when the data entries are very corrupted. In this paper, we wish to analyze the performance of these algorithms on two applications.First we will look at the problem of foreground background separation and then we will look into the problem of image inpainting. We have implemented different algorithms and analyzed their runtime and other parameters.

Index Terms— PCA, Gradient Descent, Impainting

1. INTRODUCTION

A lot of applications in the field of machine learning, data mining, computer vision require us to reduce the dimensionality of the given data. It is necessary as it is very challenging to work on high dimensional data. Principle Component Analysis is a standard procedure to undertake this task via eigen decomposition of the given data matrix. It successively finds the orthogonal principal components to capture maximum variance in the dataset. But, it has its fair share of limitations. This process is sensitive to presence of outliers and performs poorly when the data is noisy and corrupted. This calls for the need of a more robust method of undertaking these tasks which is known as Robust Principle Component Analysis(RPCA). Given a matrix with highly corrupted data, RPCA can recover the low rank component from the data. A solution to the above problem which would be provably correct and scalable could have an impact on todays data-intensive scientific discovery. Given a data matrix $M \in \mathcal{R}^{m\times n}$. RPCA aims to decompose this into two matrices L and S where $M = L+S$. Matrix L is the low rank part of the matrix and S is the sparse component. The sparse component can have sufficiently high and arbitrary values.

The generic problem of RPCA can be formulated in the following manner:

$$\min_{L,S} rank(L) + \lambda||S||_0 \qquad (1)$$

where λ is a weight parameter It has a varied number of application in a number of domains. It can be used for foreground-background videos. For this problem, the background is considered to be the low rank component assuming that it remains similar for a small video and the foreground is considered as the sparse corruption. Performing RPCA will lead us to extract these two components from the original matrix. It can also be used for Image Inpainting. The original underlying image can be reconstructed back from the corrupted image using RPCA. Here, the corrupting noise is considered as the sparse part and the underlying clean image as the low rank component. It has been used in certain Natural Language Processing Tasks for characterization of documents. Common words in the document can be formulated as being the low rank component and few uncommon words as the sparse part.

2. RELATED WORKS

Several convex as well as non-convex algorithms have been proposed to solve this problem of RPCA. Especially the convex approach of RPCA has been thoroughly studied [3]. If the location of sparse entries of matrix S is uniformly distributed and certain conditions are satisfied on the rank of low rank matrix L, L and S can be recovered with very high probability. Convex approaches usually have relatively high complexity due to the need of solving (partial) SVD of large matrices.

Cai et. al.[6] solves a relaxed version of the original RPCA problem. This method consists of relaxing the rank constraint $rank(L)$ in the optimization problem to the nuclear norm L_*.

Koh et al. [7] solves a nuclear norm regularized linear least squares problem. This regularized problem is a special case of an unconstrained non-smooth convex optimization problem, in which the objective function is the sum of a convex smooth function with Lipschitz continuous gradient and a convex function on a set of matrices. They propose an accelerated proximal gradient algorithm that terminates in $\mathcal{O}(1/\sqrt{\epsilon})$ iterations with an ϵ -optimal solution.

A fast iterative shrinkage-thresholding algorithm (FISTA) proposed by Beck et al.[8] have significantly better global convergence rate.

It is noteworthy that the convex relaxation may not be a good approximation for this problem in real-world applications. Sun et al.[9] present a novel non- convex formulation for the RPCA problem using the capped trace norm and the capped l_1-norm. For solving the non-convex problem, they propose a greedy based approach and a difference of convex functions based framework to solve the problem.

A number of other non convex approaches ranging from Alternating Projections(AltProj) to Fast factorization based approaches have been proposed with better complexity. This problem is still being studied and efforts are being made for better recovery and improving complexity.

3. IMPLEMENTED ALGORITHMS

3.1. Robust PCA using alternating projections

Netrapalli et al. [1] proposed an alternating update technique for obtaining the low-rank and the sparse component respectively.We have,

$$M = L + S, M \in \mathcal{R}^{m \times n} \tag{2}$$

The algorithm formulates the PCA problem as that of finding a matrix L of desired rank r such that it belongs to the intersection of the following two sets:

$$\mathcal{L} = \{ \text{ set of all } r \text{ rank matrices} \}$$
$$\mathcal{S}_M = \{ \; M - S, \text{where } S \text{ is a sparse matrix } \}$$

The algorithm runs for r stages,where r is the desired rank and each stage consists of certain number of iterations.In each iteration, it obtains L by projecting $M - S$ onto the set of low-rank matrices and then it updates S by projecting $M - L$ onto the set of sparse matrices.It can be observed that these sets are non-convex,but projections onto them can be done efficiently. To obtain projection onto the set of k rank matrices,singular value decomposition(SVD) is used.Let,

$$A = U\Sigma V^T \tag{3}$$

then,

$$P_k(A) = [u_1 u_2 ... u_k] diag(\sigma_1, \sigma_2, ..., \sigma_k)[v_1 v_2 ... v_k]^T \tag{4}$$

is the best k rank approximation for matrix A where $1 \leq k \leq rank(A)$.
The projection onto the set of sparse matrices is done by Hard-thresholding on $M - L$ which is described as follows:

$$(HT_\tau(M-L))_{ij} = \begin{cases} 0 & |(M-L)_{ij}| < \tau \\ (M-L)_{ij} & otherwise \end{cases} \tag{5}$$

Initialization: The low-rank component is initialized as L=0.S is initialized via hard-thresholding on M to remove very large entries from it.This threshold is $\tau = \beta\sigma_1(M)$.
β is the only significant parameter of this algorithm and it represents the "spikiness" of the low-rank component.If L is expected to have "spiky" elements,then higher values of β should be used.$\beta = 1/\sqrt{n}$ has been used in [1].

Main Loop: In the k^{th} stage where $1 \leq k \leq r$ and r is the desired rank,the algorithm runs for a fixed number of iterations T, where

$$T = 10log(n\beta log(||M - S^{(0)}||_2)/\epsilon) \tag{6}$$

where ϵ is the reconstruction error bound,

$$||M - S - L||_F \leq \epsilon^2 \tag{7}$$

.In each iteration t,the algorithm alternates between rank-k projection and hard thresholding.The threshold is updated in each iteration as

$$\tau_t = \beta(\sigma_{k+1}(M - S^{(t)}) + \frac{1}{2^t}\sigma_k(M - S^{(t)})) \tag{8}$$

After T iteration,in the k^{th} stage we check whether the remaining part has significant norm or not by the following condition

$$\beta\sigma_{k+1}(L^{(T)}) < \frac{\epsilon}{2n} \tag{9}$$

If this condition is satisfied,$L^{(T)}, S^{(T)}$ are returned otherwise the process is continued with

$$S^{(0)} = S^{(T)} \tag{10}$$

The overall time-complexity of the algorithm is $\mathcal{O}(r^2mn)$. The pseudo-code and rest of the details can be found in [1].

3.2. Robust PCA using IALM method

Given a matrix M such that

$$M = L + S \tag{11}$$

[5] formulates the RPCA problem as follows:

$$\{\hat{L}, \hat{S}\} = argmin||L||_* + \lambda||S||_1, M = L + S \tag{12}$$

Hence,the lagrangian after including the quadratic penalty term is:

$$\mathcal{L}(L, S, Y, \mu) = ||L||_* + \lambda||S||_1 + \langle Y, M - L - S\rangle + \frac{\mu}{2}||M - L - S||_F^2 \tag{13}$$

The IALM approach solves this convex program by utilising an alternating optimization technique to update L and S.
Initialization:Y is initialized as follows:

$$Y^0 = \frac{sgn(M)}{J(sgn(M))}, J(Y) = max(||Y||_2, \lambda^{-1}||Y||_\infty) \tag{14}$$

μ_0,S and ρ,which is used to update μ are set as:

$$\mu_0 = \frac{1.25}{||M||_2}, S = 0, \rho = 1.6 \tag{15}$$

Main loop: In each iteration, L is updated as follows:

$$L_{k+1} = \arg\max_{L} \mathcal{L}(L, S_k, Y_k, \mu) \tag{16}$$

This is done as follows:

$$(U, \Sigma, V) = \textbf{SVD}(M - S_k + \frac{Y_k}{\mu_k}) \tag{17}$$

$$A_{k+1} = U\mathcal{S}_{\mu_k^{-1}}[\Sigma]V^T \tag{18}$$

Here, $\mathcal{S}_\epsilon[x]$ is the soft-thresholding(shrinkage) operator:

$$\mathcal{S}_\epsilon[x] = \begin{cases} x - \epsilon & x > \epsilon \\ x + \epsilon & x < -\epsilon \\ 0 & otherwise \end{cases} \tag{19}$$

S is updated as follows:

$$S_{k+1} = \arg\max_{S} \mathcal{L}(L_{k+1}, S, Y_k, \mu_k) \tag{20}$$

Which is done as follows:

$$S_{k+1} = \mathcal{S}_{\lambda\mu_k^{-1}}[M - L_{k+1} + \frac{Y_k}{\mu_k}] \tag{21}$$

Y and μ are updated as follows:

$$Y_{K+1} = Y_k + \mu_k(M - L_{k+1} - S_{k+1}), \mu_{k+1} = \rho\mu_k \tag{22}$$

The main loop is terminated and L_{k+1},S_{k+1} are returned if the following convergence criterion is met:

$$||M - L_{k+1} - S_{k+1}||_F < 10^{-7}||M||_F \tag{23}$$

The pseudo-code and rest of the details of the algorithm can be found at [5]

3.3. Robust PCA via Gradient Descent

This approach is an non-convex method based on the projected gradient descent on factorized space. It is assumed that deterministic corruptions are spread out and there is an upper limit on their number in each row and column.

We consider the problem of observing the matrix $Y \in \mathbf{R}^{d_1 \times d_2}$ such that $Y^* = M^* + S^*$ where M^* is the low rank part and S^* has sparse support with corrupted entries[2].

$$Y_{i,j} = \begin{cases} (M * + S*)_{i,j} & \text{with probability p} \\ * & \text{otherwise} \end{cases} \tag{24}$$

It is assumed that entries of $Y_{i,j}$ are revealed independently with probability p. M* cannot be low rank and sparse at the same time.

An iterative gradient based method is used to solve this problem. Input is an observed matrix Y with desired rank r, corruption factor α.[2]

Initialize S_{init} with a sparse estimation of matrix Y defined as $\mathcal{T}_\alpha[Y]$. Top α-fraction entries in Y corresponding to each row and column are kept intact and rest other entries are made to 0 to get the sparse estimator and matrix S_{init}. Perform SVD operation on $Y - S_{init}$ to get a rough estimate of the low rank part through its decomposition U_0 and V_0.

Once we have U_0 and V_0 for iteration $t = 0$ we apply gradient based iteration to get the final output. For iteration t, update S as

$$S_t \leftarrow \mathcal{T}_\alpha[Y - U_t V_t^T] \tag{25}$$

To get U_{t+1} and V_{t+1}, perform a projected gradient descent on low rank factorized space.

$$U_{t+1} \leftarrow \Pi_{\mathcal{U}}(U_t - \eta\Delta_U \mathcal{L}(U_t, V_t; S_t) - \frac{1}{2}\eta U_t(U_t^T U_t - V_t^T V_t)) \tag{26}$$

$$V_{t+1} \leftarrow \Pi_{\mathcal{V}}(V_t - \eta\Delta_V \mathcal{L}(U_t, V_t; S_t) - \frac{1}{2}\eta V_t(V_t^T V_t - U_t^T U_t)) \tag{27}$$

where $\mathcal{L}(U_t, V_t; S_t)$ is the loss function defined as the frobenius norm of a matrix.

$$\mathcal{L}(U, V; S) = \frac{1}{2}||UV_T + SY||_F^2, \tag{28}$$

We iterate through the loop until and unless we don't converge. At the end of this algorithm we have U_T and V_T which are the final values for these matrices. Obtain the low rank part M by setting it equal to $U_T V_T^T$

Projection on set $\mathcal{U}$ is done so as to preserve μ coherent structure of matrix M. It ensures that M is not too sparse. Hence, a constraint is put on the row norms of U and V produced during the iterations.

Even though the algorithm is convex, it seems to have linear convergence guarantees. If all the initialization are performed properly, the algorithm can be solved with the complexity $\mathcal{O}(rd^2 \log(1/\epsilon))$ with the robustness value α of the order of $\mathcal{O}(1/(\mu r^{1.5}))$ [2]. This algorithms outperforms other best know algorithms by a factor of r.

3.4. Non-convex rank approximation based Robust PCA

[3] formulates the problem of RPCA as follows:

$$\min_{L,S} ||L||_\gamma + \lambda||S||_l \;\; s.t \;\; M = L + S \tag{29}$$

where $rank(L)$ is approximated as the $||.||_\gamma$ and termed as γ-norm,and $||.||_l$ is a proper l-norm.The γ-norm is defined as follows:

$$||L||_\gamma = \sum_i \frac{(1+\gamma)\sigma_i(L)}{\gamma + \sigma_i(L)}, \;\; \gamma > 0 \tag{30}$$

Further,it can be easily observed that,

$$\lim_{\gamma \to 0} \; ||L||_\gamma = rank(L) \tag{31}$$

$$\lim_{\gamma \to \infty} ||L||_\gamma = ||L||_* \quad (32)$$

Further,[3] suggests that $\gamma = 0.01$ closely approximates the true rank. The optimization problem after obtaining the lagrangian with Y being the lagrange multiplier and adding quadratic penalty term becomes:

$$\mathcal{L} = ||L||_\gamma + \lambda||S||_l + \langle Y, L+S-M\rangle + \frac{\mu}{2}||M-L-S||_F \quad (33)$$

Where the inner product of two matrices is $\langle A, B\rangle = tr(A^T B)$ and μ is the regularization parameter. Now,an alternating optimization technique is used to update L,S.
Initialization: $L \leftarrow 0$,$S \leftarrow 0$
Main Loop: In t^{th} iteration of the main loop L^t is updated as follows:

$$L^t = \underset{L}{argmin}\ \mathcal{L}(L, S^{t-1}, Y^{t-1}, \mu^{t-1}) \quad (34)$$

follwed by upadating S^t as follows:

$$S^t = \underset{S}{argmin}\ \mathcal{L}(L^t, S, Y^{t-1}, \mu^{t-1}) \quad (35)$$

Y^t and μ^t are updated as follows:

$$Y^t = Y^{t-1} + \mu^{t-1}(L^t + S^t - M) \quad (36)$$

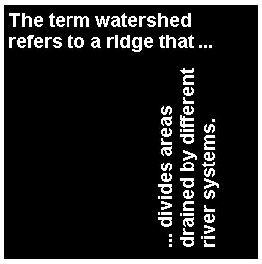
$$\mu^t = \rho\mu^{t-1} \quad (37)$$

And the convergence criterion used is as follows:

$$\frac{||M - L^t - S^t||_F}{||M||_F} \leq \epsilon \quad (38)$$

4. DATASET

For the foreground-background separation task, we will use the readily available Escalator video which is widely used for for comparing RPCA algorithms.We take 196 frames of this video each of size 160×130 for our experiments.



The term watershed
refers to a ridge that ...

... divides areas
drained by different
river systems.

Fig. 1: Corrupting Image for Image Impainting Task

For the task of image impainting, we need to corrupt the images with noise. We use the already available image of moon from matlab for this task. We corrupt this image with a text image masked over it.

5. APPLICATIONS

5.1. Video Foreground-Background Separation

In motion videos where foreground objects are moving with respect to a constant background, this method can be very helpful. Robust PCA models background as the low rank component and the foreground as the sparse corruptions. We have implemented on videos whose frames are converted to grayscale. It can similarly be extended to colored videos as well.

We apply all the four algorithms for performing this task.

5.1.1. RPCA using Alternating Projections

For this approach, we have obtained the best rank-1 approximation of the low rank component. And,we have set $\beta = 1/\sqrt{n}$,$\epsilon = 10^{-3}$,where n is the number of columns of the observed matrix.

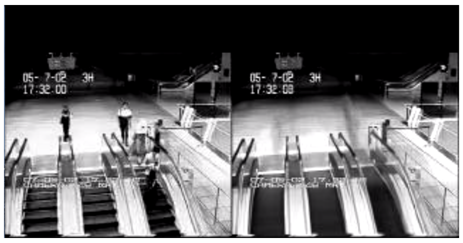


Fig. 2: Screenshot: Left frame is of original video; right frame is the background after applying Alternate Projection algorithm

5.1.2. RPCA using IALM

In algorithm,we use the parameters with the same values as mentioned in section 3.2 and $\lambda = \frac{1}{\sqrt{m}}$,where m is the number of rows of the observed matrix.

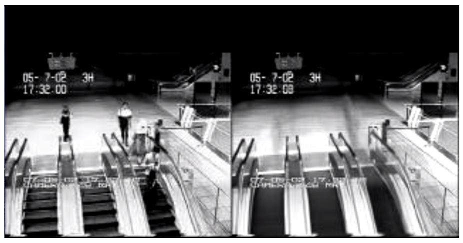


Fig. 3: Screenshot: Left frame is from original video; right frame is the background after applying IALM algorithm

5.1.3. Fast RPCA via gradient descent

In this algorithm,we find a rank-1 approximation of the observation matrix using $\alpha = 0.15$,$tolerance = 2 \times 10^{-4}$,$\eta = 0.5$,$\gamma = 1.5$

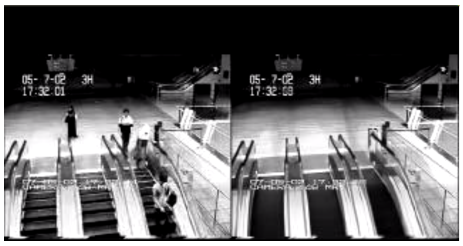


Fig. 4: Screenshot: Left frame is of original video; right frame is the background after applying Fast gradient based method

5.1.4. *Non-convex rank approximation based RPCA*

The following parameter values are used $\mu_0 = 0.005$,$\rho = 1.1$, $\lambda = \frac{1}{\sqrt{\max m,n}}$.

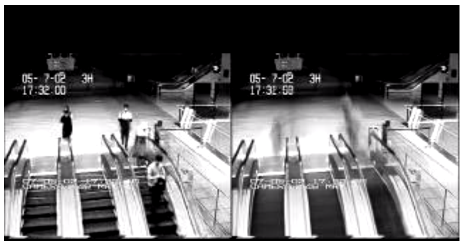


Fig. 5: Screenshot: Left frame is of original video; right frame is the background after applying Fast gradient based method

5.2. Image Inpainting

Image Inpainting is a task where a corrupted part of an image has to be recovered. It is a heavily researched task. Various methods ranging from Orthogonal Matching Pursuit to IRLS are used for solving this problem.

To solve this problem, overlapping image patches of a fixed window size are created and flattened to generate a matrix. Different Robust PCA algorithms are then used on this matrix to recover the low rank part from it. Here, the unknown pixels which have been masked by noise are corruptions and constitute the sparse component. The underlying clean image is the low rank part which has to be recovered.
In this application,we have implemented only the first 3 algorithms.

5.2.1. *RPCA using Alternating Projections*

For this approach,we have obtained the best rank-1 approximation of the low rank component.And,we have set $\beta = 1/\sqrt{n}$,$\epsilon = 10^{-3}$,where n is the number of columns of the observed matrix.

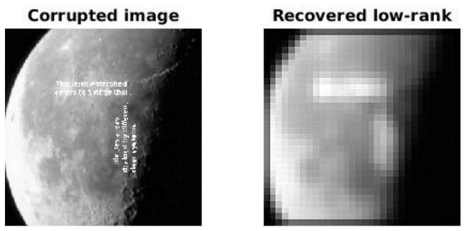


Fig. 6: Inpainting Using Alternating Projection

5.2.2. *RPCA using IALM*

In algorithm,we use the parameters with the same values as mentioned in section 3.2 and $\lambda = \frac{1}{\sqrt{m}}$,where m is the number of rows of the observed matrix.

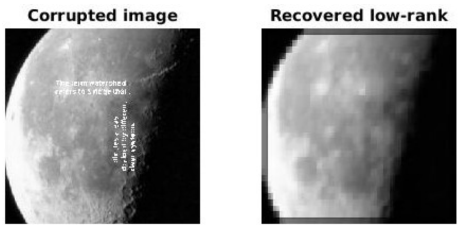


Fig. 7: Inpainting Using IALM algorithm

5.2.3. *Fast RPCA via gradient descent*

In this algorithm,we find a rank-1 approximation of the observation matrix using $\alpha = 0.15$, $tolerance = 2 \times 10^{-4}$, $\eta = 0.5$,$\gamma = 1.5$

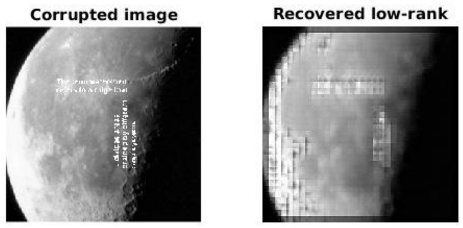


Fig. 8: Inpainting using Fast gradient based method

6. RESULTS

6.1. Background-foreground seperation

Performance comparison			
Algorithms	Rank	$\frac{\|\|M-L-S\|\|_F}{\|\|M\|\|_F}$	run-time
AltProj	1	0	128.6174s
IALM	6	3.0958e-13	49.5730s
Fast RPCA	2	0.1232	27.2s
Non-convex RPCA	2	3.44e-7	34.2688s

6.2. Image inpainting

Performance comparison			
Algorithms	Rank	$\frac{\|\|M-L-S\|\|_F}{\|\|M\|\|_F}$	run-time
AltProj	1	0	3.9246s
IALM	6	2.497910e-10	4.8690s
Fast RPCA	6	0.121	0.98s

6.3. Inferences

In terms of visual quality of the extracted low-rank component in the foreground-background task,all algorithms give similar quality.
However,in the image inpainting task IALM seems to give better results than the other two algorithms.

7. ACKNOWLEDGMENTS

While implementing [2],[3],[5], we have taken help of some part of the codes that were provided by the authors for reference. PROPACK package was used directly. However,we implemented [1] from scratch.

8. FUTURE WORK

In the video background-foreground application,we took only 196 frames for our experiments due to memory and speed constraints.Also,we were unable to implement [4] due to difficulties in implementing the proximal operator used in it.So,in future one can extend this work and compare these algorithms on more number of frames and try to implement 2D-RPCA.

9. REFERENCES

[1] Praneeth Netrapalli, U N Niranjan, Sujay Sanghavi, Animashree Anandkumar, Prateek Jain ”Provable Non-convex Robust PCA” Advances in Neural Information Processing Systems 2014,

[2] Xinyang Yi, Dohyung Park, Yudong Chen, Constantine Caramanis ”Fast algorithms for robust PCA via gradient descent” Advances in Neural Information Processing Systems, 2016

[3] Zhao Kang, Qiang Cheng, Chong Peng. ”Robust PCA via Nonconvex Rank Approximation” Proceedings of IEEE ICDM 2015

[4] Yipeng Sun, Yang Li. ”Robust 2D Principal Component Analysis: A Structured Sparsity Regularized Approach” IEEE Transactions on Image Processing 2015

[5] Z Lin, M Chen, Y Ma ”The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices” arXiv preprint arXiv:1009.5055, 2010

[6] J.-F. Cai, E. J. Cands, and Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization, vol. 20, no. 4, pp. 19561982, 2010.

[7] K.-C. Toh and S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, Pacific Journal of Optimization, vol. 6, no. 615-640, p. 15, 2010.

[8] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences, vol. 2, no. 1, pp. 183202, 2009

[9] Q. Sun, S. Xiang, and J. Ye, Robust principal component analysis via capped norms, in Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2013, pp. 311319.

[10] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Rank- sparsity incoherence for matrix decomposition. In SIAM Journal on Optimization 21.2 (2011), pp. 572596.