Applications of Low Rank Matrix Completion

Project Report for CS754: Advanced Image Processing

Harsh Shah
Dept. of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
200050049@iitb.ac.in

Pranava Singhal
Dept. of Electrical Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
200070057@iitb.ac.in

I. Experiments

1) Video Denoising
We implemented this paper, which deals with methods of low rank matrix completion in order to reduce spike, Poisson and Gaussian noise from videos. Videos were taken from xiph.org. We added varying levels of **mixed** noise to the videos and then compared the denoising algorithm against other popular techniques. The algorithm briefly consists of the following steps: adaptive median filtering to remove spike noise and identify missing pixels (temporary filtering), matching similar patches across all frames based on mean absolute difference, further removal of some noisy pixels, recovering denoised patches using low rank matrix completion on the patch matrix using fixed point minimisation and finally, concatenating denoised overlapping patches to recover the denoised video. The average PSNR of the first 4 denoised frames comes out to be 22.96 dB.
2) Image Inpainting
Similar to the above paper, we implemented image inpainting on the famous Lenna image. 5x5 blocks were removed (instead of uniformly distributed missing entries) and then the image was reconstructed using the same algorithm as above.
3) Reweighted Nuclear Norm Minimisation
We also implemented a reweighted modification of nuclear norm minimisation as done here and compared it with the unweighted algorithm in our main paper.
4) Variation of results with μ
One of the parameters for the matrix completion algorithm is μ which occurs in the cost function being minimised $\min Q \, \frac{1}{2}||Q|_\Omega - P|_\Omega||_F^2 + \mu||Q||_*$. While the paper mentions a formula for μ based on heuristic arguments we found that often a better value of μ can be found by searching across different values of μ (similar to cross-validation). The algorithm's performance is very sensitive to this parameter.

II. Results

1) Testing the fixed point minimisation algorithm on small matrices

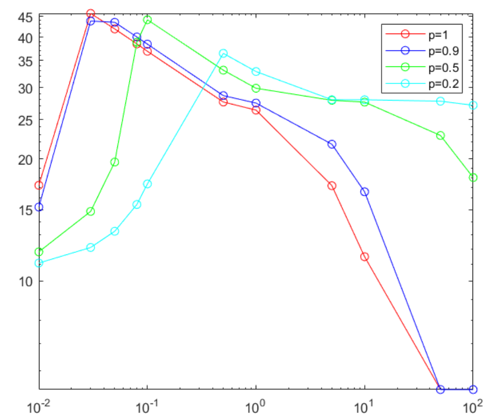

Fig. 1. PSNR for matrix recovery, varying p and μ

Small low rank matrices are generated by multiplying two random matrices and a projection mask Ω is generated. We evaluate the performance of recovery algorithms for different Schatten-p norms (p =1 is the nuclear norm, p¡1 are non-convex approximations of rank, solved by iterative reweighting) and for different values of μ. In particular we take 100x100 matrices with rank = 20 and p = 70 percent observed entries. $\tau = 1.5$, $\mu = \mu_0 \sigma$ where σ is varied from 0.01 to 100 on a log scale. We plot PSNR and RMSE for recovered matrices in each case for different μ values and p norms.
It can be seen that $p < 1$ gives a better approximation of rank and thus weighted singular value thresholding performs better at low rank matrix recovery. Optimal recovery happens at a particular μ and error increases for too large or too small μ.
2) Video Denoising

- A 64x64 region from each RGB frame was taken for computational speed. Gaussian noise variance σ was fixed at 10, Poisson noise variance scaling factor $\kappa = 5$, spike noise with probability $s = 0.3$ with equal probability across all channels R,G,B.

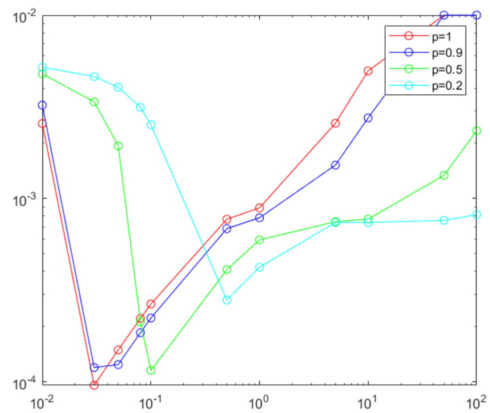

Fig. 2. RMSE for matrix recovery, varying p and μ

Fig. 3. Original video frame

First median filtering is applied to all frames to remove spike noise. This filtering is temporary and is only used for patch matching. The resulting image is displayed below.
Each frame is used to generate overlapping patches with stride = 4. After running the patch matching algorithm and singular value thresholding (fixed point minimisation) we recover the denoised image displayed below. Blocking artifacts are visible due to the stride of 4. The image appears blurred compared to the original.

3) Inpainting

- We took a 100x100 portion of the Lenna image, some pixels were removed in groups of 5x5 blocks randomly distributed (the observed pixels were therefore not uniformly distributed). From the image and the observation mask, overlapping patches were extracted, recovered using fixed point minimisation and then the image is reconstructed. The experiment was performed for both stride = 1 (dense patches) and stride = 4, with all other parameters same. Reconstruction preserves edges better in the stride = 1 case, but the intensity of reconstructed pixels in missing locations is lower which may be due to the sensitivity of reconstruction to μ.
- We tried a similar experiment on this image and performed reconstructions for different values of μ and for both p =1 and p=0.5 (reweighted nuclear norm). Just like the case of video denoising, reconstruction quality (PSNR, RMSE) vary significantly with μ. p=0.5 gives better reconstruction quality. Stride was taken as 4. While low μ results in almost zero intensity at missing pixels (low weightage to

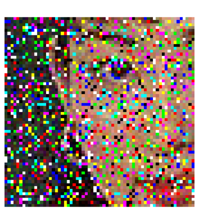

Fig. 4. Frame after adding noise

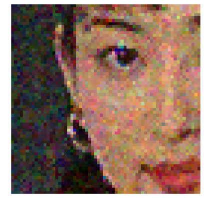

Fig. 5. Frame after Adaptive Median Filtering

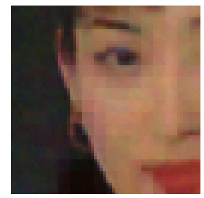

Fig. 6. Reconstructed frame using low rank matrix completion

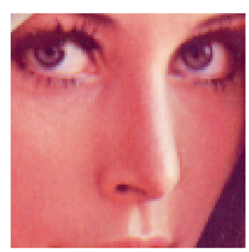

Fig. 7. Original image

Fig. 8. Image with certain blocks removed

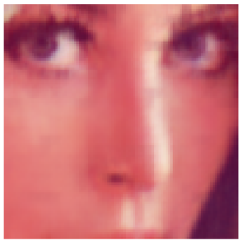

Fig. 9. Reconstructed with overlapping patches, stride = 4

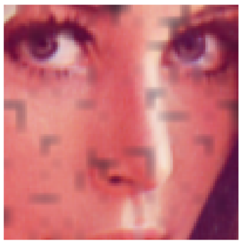

Fig. 10. Reconstructed with overlapping patches, stride = 1 (dense)

nuclear norm term in cost function), high μ results in excessive shrinkage making the image nearly zero (dark) in both cases.

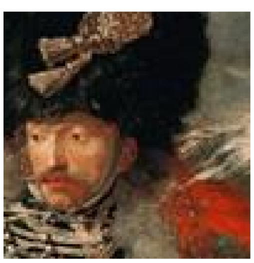

Fig. 11. Original

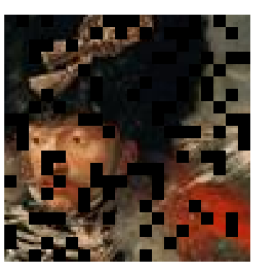

Fig. 12. Corrupted with 5x5 blocks

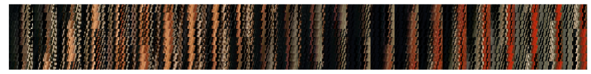

Fig. 13. Patch Matrix 64x576

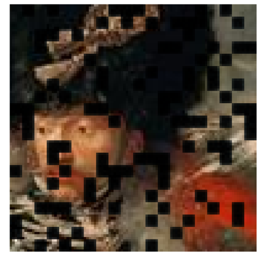

Fig. 14. Reconstructed with $\mu = 0.01$, p=1

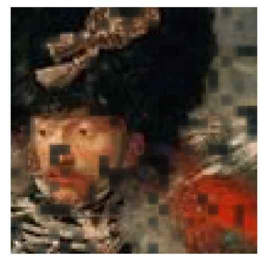

Fig. 15. Reconstructed with $\mu = 0.1$, p=1

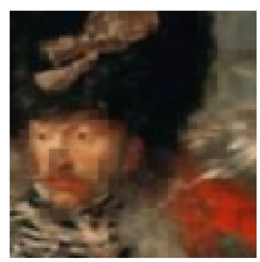

Fig. 16. Reconstructed with $\mu = 1$, p=1

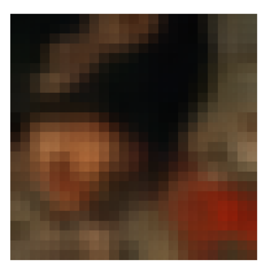

Fig. 17. Reconstructed with $\mu = 10$, p=1

Fig. 18. Reconstructed with $\mu = 100$, p=1

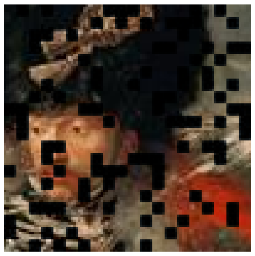

Fig. 19. Reconstructed with $\mu = 0.01$, p=0.5

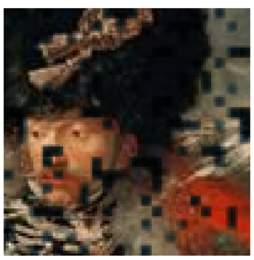

Fig. 20. Reconstructed with $\mu = 0.1$, p=0.5

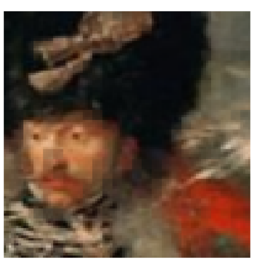

Fig. 21. Reconstructed with $\mu = 1$, p=0.5

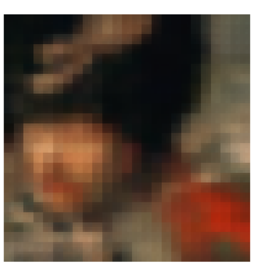

Fig. 22. Reconstructed with $\mu = 10$, p=0.5

Fig. 23. Reconstructed with $\mu = 100$, p=0.5

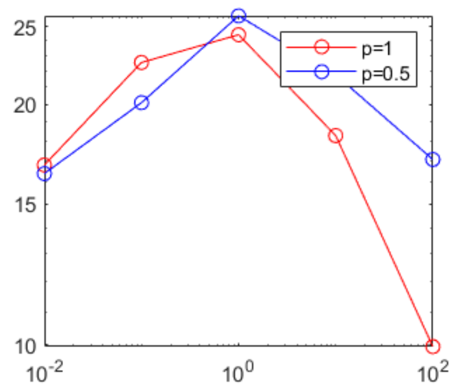

Fig. 24. PSNR vs μ

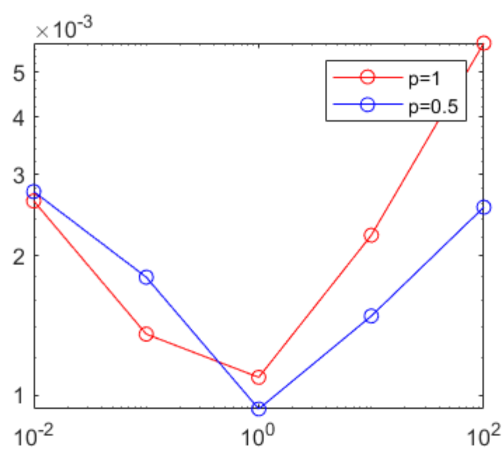

Fig. 25. RMSE vs μ