I. Robust PCA with known rank: a block coordinate descent approach

A. Motivation

In some application, we normally have some prior information about the low rank component of an observed matrix. For examples, in computer vision, if we are to extract the background from different frames of a video, then it is natural to consider each video frame as a long vector. The background that we are recovering is then a long vector, which is of rank 1. Therefore, it is natural to see if we can utilize this additional rank information to derive faster algorithms while retaining the performance guarantee.

B. Equivalence formulation of Robust PCA with rank information

We first show~~s~~ that in the robust PCA framework, the same probability guarantee will still hold when ~~extra information is given~~. Then we derive a block-coordinate descent formulation for the case when rank information is known.

Proposition 1. *Let $M = L_0 + S_0$, $rank(L_0) \leq r$ and (L_0, S_0) satisfy the Robust PCA assumptions. Then with high probability, the following problems are equivalent:*

$$
\begin{aligned}
J_1 \quad &= \min\nolimits_{L,S} \quad ||L||_* + \lambda ||S||_1 \\
&\quad s.t. \qquad M = L + S
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
J_2 \quad &= \min\nolimits_{L,S} \quad ||L||_* + \lambda ||S||_1 \\
&\quad s.t. \qquad M = L + S \\
&\qquad\qquad\quad and\ T(L, S)\ holds
\end{aligned}
\tag{2}
$$

where $T(L, S)$ are conditions on (L, S) that (L_0, S_0) also satisfies.

Proof: We use subscript to denote the optimizer for J_1 and J_2 respectively. With high probability, $(L_1, S_1) = (L_0, S_0)$. Now, over the event of $(L_1, S_1) = (L_0, S_0)$, (L_1, S_1) is also a feasible solution for (2). Since $J_1 \leq J_2$ always, now (L_1, S_1) achieve the bound for (2). Note that it should also be unique because otherwise it would contradict with the recovery of (1). ■

Now we apply Proposition (1) to the case when the rank information is known and derive a block coordinate descent framework for future algorithm design.

$$
\begin{aligned}
J_3 \quad &= \quad \min_{M=L+S, rank(L) \leq r} ||L||_* + \lambda ||S||_1 \\
&= \quad \min_{L,S,\mu_i,p_i q_i} ||L||_* + \lambda ||S||_1 \\
s.t. \quad & S = M - L, L = \sum_{i=1}^{r} \mu_i p_i q_i^T \\
& ||p_i||_2 = 1, ||q_i||_2 = 1, \mu_i \geq 0 \\
&= \quad \min_{\mu_i, p_i q_i} \sum_{i=1}^{r} \mu_i + \lambda ||M - \sum_{i=1}^{r} \mu_i p_i q_i^T ||_1 \\
s.t. \quad & ||p_i||_2 = 1, ||q_i||_2 = 1, \mu_i \geq 0 \\
&= \quad \min_{\mu_i, p_i q_i} \sum_{i=1}^{r} |\mu_i - 0| + \lambda ||M - \sum_{i=1}^{r} \mu_i p_i q_i^T ||_1 \\
s.t. \quad & ||p_i||_2 = 1, ||q_i||_2 = 1, \mu_i \geq 0
\end{aligned}
\tag{3}
$$

Note that this formulation allows us to optimize over $\mu_i, p_i q_i$ sequentially and for each step of optimization, the problem is a weighted median problem where efficient algorithm is known. And by the Proposition (1), we know that the formulation of (3) can recover the orginal (L_0, S_0) with high probability.

C. Simplification using L_1 heuristic

1) Introduction: Recall that the nuclear norm introduced in the PCP scheme is resulted because we would like to extract the low rank component from gross random noise. Nuclear norm is used because it is a heuristic for penalizing high rank matrix. Now, we consider the case when we have extra information/guess from the data set that we know precisely what the rank of the matrix are. Therefore, it is natural to introduce the following heuristics.

$$E^* = \min_{\{p_j\}\{q_j\},1\le j\le r} ||M - \sum_{j=1}^{r} p_j q_j^T||_1 \tag{4}$$

2) Performance guarantee for the L_1 heuristic: For this new heuristic, we provide some performance guarantee for the case when the noise is bounded. One is a result for deterministic case and the other is for the random case. They are as follows.

Proposition 2. *Let $M = S + \sum_{i=1}^{r} p_i q_i^T$, $\frac{2}{\epsilon}||S||_1 \le ||\sum_{i=1}^{r} p_i q_i^T||_1$, then, for the recovered $(\hat{L}, \hat{S})$ from (4) , it will satisfy*

$$\frac{||\sum_{i=1}^{r} p_i q_i^T - \hat{L}||_1}{||\sum_{i=1}^{r} p_i q_i^T||_1} \le \epsilon$$

Proof: Assume not, then

$$\begin{aligned} ||S||_1 &\ge ||\sum_{i=1}^{r} p_i q_i^T + S - \hat{L}||_1 \\ &\ge ||\sum_{i=1}^{r} p_i q_i^T - \hat{L}||_1 - ||S||_1 \\ &> \epsilon||\sum_{i=1}^{r} p_i q_i^T||_1 - ||S||_1 \end{aligned}$$

■

This gives a contradition, which is,

$$\frac{2}{\epsilon}||S||_1 > ||\sum_{i=1}^{r} p_i q_i^T||$$

Proposition 3. *Let $M = \sum_{i=1}^{r} p_i q_i^T + S$, where $S_{i,j} \sim Uniform(-x_s, x_s)$, $(p_i)_j \sim Uniform(-x_p, x_p)$, $(q_i)_j \sim Uniform(-x_q, x_q)$ all of the random variables being independent. With $|S| = k$ such that $\lim_{n\to\infty} \frac{k^2}{n}$, then we have,*

$$\lim_{n\to\infty} P\left(\frac{||\sum_{i=1}^{r} p_i q_i^T - \hat{L}||_1}{||\sum_{i=1}^{r} p_i q_i^T||_1} > \epsilon\right) = 0$$

Proof: Let E be the error event that $\frac{||\sum_{i=1}^{r} p_i q_i^T - \hat{L}||_1}{||\sum_{i=1}^{r} p_i q_i^T||_1} > \epsilon$. If error occurs,

$$\begin{aligned} kx_s &\ge ||\sum_{i=1}^{r} p_i q_i^T + S - \hat{L}||_1 \\ &\ge ||\sum_{i=1}^{r} p_i q_i^T - \hat{L}||_1 - ||S||_1 \\ &\ge \epsilon||\sum_{i=1}^{r} p_i q_i^T||_1 - kx_s \\ &\ge \epsilon\sqrt{\sum_{l_1 l_2}(\sum_{i=1}^{r}(p_i)_{l_1}(q_i)_{l_2})^2} - \cancel{k_s} \end{aligned}$$

Thus,

$$\begin{aligned} & Pr(E) \\ \le\ & Pr((\tfrac{2kx_s}{\epsilon})^2 \ge \textstyle\sum_{l_1 l_2}(\sum_{i=1}^{r}(p_i)_{l_1}(q_i)_{l_2})^2) \\ \le\ & Pr((\tfrac{2kx_s}{\epsilon})^2 \ge \textstyle\sum_{l_1=1}^{n}(\sum_{i=1}^{r}(p_i)_{l_1}(q_i)_{l_1})^2) \\ =\ & Pr(\tfrac{1}{n}(\tfrac{2kx_s}{\epsilon})^2 \ge \tfrac{1}{n}\textstyle\sum_{l_1=1}^{n}(\sum_{i=1}^{r}(p_i)_{l_1}(q_i)_{l_1})^2) \end{aligned}$$

Moreover, as $E(\sum_{i=1}^{r}(p_i)_{l_1}(q_i)_{l_1})^2) = \frac{r}{3}x_p^2x_q^2$, by law of large number, $\frac{1}{n}\sum_{l_1=1}^{n}(\sum_{i=1}^{r}(p_i)_{l_1}(q_i)_{l_1})^2) \to \frac{r}{3}x_p^2x_q^2$. Thus, since $\frac{1}{n}(\frac{2kx_s}{\epsilon})^2 \to 0$. This gives $Pr(E) \to 0$ as $n \to \infty$. ■

However, we know that the L_1 heuristic cannot work well in the case of unbounded noise. In particular, we consider the following example.

Say , for $n = 100$, $L_0 = [\begin{matrix} 1 \\ 1 \\ . \\ . \\ 1 \end{matrix}][\begin{matrix} 1 & 1 & . & . & 1 \end{matrix}], S_0 = [\begin{matrix} 10^9 & 0 & 0 & ... & 0 \\ 0 & 0 & 0 & ... & 0 \\ . & . & . & ... & . \\ . & . & . & ... & . \\ 0 & 0 & 0 & ... & 0 \end{matrix}]$, then by the L_1 heuristics, we would get $\hat{L_0} \sim [\begin{matrix} 10^9 \\ 1 \\ . \\ . \\ 1 \end{matrix}][\begin{matrix} 1 & 10^{-9} & . & . & 10^{-9} \end{matrix}], \hat{S_0} \sim [\begin{matrix} 1 & 0 & 0 & ... & 0 \\ 0 & 1 & 1 & ... & 1 \\ . & . & . & ... & . \\ . & . & . & ... & . \\ 0 & 1 & 1 & ... & 1 \end{matrix}]$, which deviates from what we expect. However, the robust PCA can recover even for this case because it penalize the nuclear norm of L.

D. *Algorithms derivation*

Note that the form for both (3) and (4) are similar. Indeed, one can generalize the method for $r = 1$ in (4) to higher dimensions for both (3) and (4). Therefore, we restrict our discussion to $r = 1$ in (4).

Let $M = (a_{i,j}) \in R^{mXn}$. We now employ the block-coordinate descent approach to solve this problem. Note that

$$\begin{aligned} \min_p \quad ||M - pq^T||_1 &= \sum_{i=1}^{m} \min_t (\sum_{j=1}^{n} |a_{i,j} - tq_j|) \\ &= \sum_{i=1}^{m} \min_t (\sum_{j=1}^{n} |q_j||t - \frac{a_{i,j}}{q_j}|) \end{aligned} \tag{5}$$

$$\begin{aligned} \min_q \quad ||M - pq^T||_1 &= \sum_{j=1}^{n} \min_t (\sum_{i=1}^{m} |a_{i,j} - tp_i|) \\ &= \sum_{j=1}^{n} \min_t (\sum_{j=1}^{n} |p_i||t - \frac{a_{i,j}}{p_i}|) \end{aligned} \tag{6}$$

And for solving the subproblem of finding

$$\min_t \sum_{k=1}^{k_0} c_i|t - d_i|$$

where $c_i \geq 0$ is basically finding the weighted median and can be done by the following method with complexity $O(k_0 \log k_0)$ mostly on sorting the sequence. We call it WMH.

Algorithm 1 WMH $(k_0, \vec{c}, \vec{d})$

1) We first sort $\vec{d}$ s.t. $d_{i_1} \leq d_{i_2} \leq ... \leq d_{i_{k_0}}$
2) We then find $k^{'}$s.t.

$$\begin{aligned} \sum_{\theta=1}^{k^{'}-1} c_{i_\theta} &\leq \sum_{\theta=k^{'}}^{k_0} c_{i_\theta} \\ \sum_{i=1}^{k^{'}} c_{i_\theta} &\geq \sum_{i=k^{'}+1}^{k_0} c_{i_\theta} \end{aligned}$$

3) We then set $t^*$to be $d_{i_{k^{'}}}$

This algorithm is optimal in finding t. This is justified by using the property of sub-differential of $|| \cdot ||_1$ and note that $0 \in \partial(\sum_{k=1}^{k_0} c_i|t^* - d_i|)$.

Now we are ready to state the power iteration method to solve the rank-1 optimization problem. We call it Poweriteration.

Algorithm 2 Poweriteration(M)

Repeat

1) $p_i \leftarrow wmh(n, abs(q), M(i,:)./q)$ for each i
2) $q_j \leftarrow wmh(n, abs(p), M(:,j)./p)$ for each j

Until stopping criterion is met

E. Sensitivity to λ

Note that in the robust PCA framework, the parameter λ is explicit stated as $\sqrt{\frac{1}{n}}$ and when the λ is too large or too small, it would significantly affect the recovery. However, in the case that the rank information is explicitly stated. The effect of λ is different. As we have seen in previous section, when the value of $\lambda$is large, it correspond to the L_1 heuristic and there is some guarantee on recovery. Therefore, it is natural to ask if we can have very small λ and have the robust PCA framework to hold. In particular, we specialize to the rank 1 case, and it turns out that it cannot be done, as demonstrated as follows.

Recall that if we directly apply robust PCA , we will get,

$$\begin{aligned}
& \min_{M=L+S, rank(L)\leq 1} ||L||_* + \lambda ||S||_1 \\
= & \min_{S=M-pq^T, L=pq^T} ||L||_* + \lambda ||S||_1 \\
= & \min_{S=M-pq^T, L=pq^T} ||pq^T||_* + \lambda ||M-pq^T||_1 \\
= & \min_{p,q:||p||_2=1} ||pq^T||_* + \lambda ||M-pq^T||_1 \\
= & \min_{p,q:||p||_2=1} ||q||_2 + \lambda ||M-pq^T||_1 \\
= & \min_{p:||p||_2=1} \min_q ||q||_2 + \lambda ||M-pq^T||_1
\end{aligned}$$

Now, for every fixed p, we consider the subproblem, if we directly apply the Robust PCA with $\lambda \leq \frac{1}{n}$,

$$\begin{aligned}
& \min_q ||q||_2 + \lambda ||M-pq^T||_1 \\
= & \min_q \max_{||u||_2\leq 1, ||V||_\infty \leq 1} u^T q + \lambda Tr(V^T(M-pq^T)) \\
= & \max_{||u||_2\leq 1, ||V||_\infty \leq 1} \min_q u^T q + \lambda Tr(V^T(M-pq^T)) \\
= & \max_{||u||_2\leq 1, ||V||_\infty \leq 1, u=\lambda V^T p} Tr(V^T M) \\
= & \max_{||\lambda V^T p||_2\leq 1, ||V||_\infty \leq 1} Tr(V^T M) \\
= & \max_{||V^T p||_2\leq n, ||V||_\infty \leq 1} Tr(V^T M)
\end{aligned}$$

Now note that, since $||p||_2 \leq 1$, we have $||V^T p||_2 \leq \sqrt{\sum_{i=1}^n \sigma_i(V^T V)} = \sqrt{Tr(V^T V)} \leq \sqrt{n^2 ||V||_\infty}$. Thus, the optimal value is

$$\begin{aligned}
& \min_{p,q:||p||_2=1} ||pq^T||_* + \lambda ||M-pq^T||_1 \\
= & \min_{p:||p||_2=1} \max_{||V||_\infty \leq 1} Tr(V^T M) \\
= & \min_{p:||p||_2=1} ||M||_1 \\
= & ||M||_1
\end{aligned}$$

And it is achieved by $pq^T = 0$ matrix, which deviates from what we expect to recover.

F. Simulation

1) Comparison between Robust PCA and L_1 heuristics: We simulate the Robust PCA scheme and L_1 heuristics using the power method. Note that in using the power iteration method for Robust PCA, we would not update μ if the value of that iteration is 0 because this will make the algorithm to converge to the wrong value(as observed from simulation, this happens quite frequently so this conditioning is needed).

In the simulation, we use randomly generated the entries of p and q as $N(0,1)$ iid. And then we randomly generate sparse matrix with sparse support uniformly distributed across the nXn matrix. And each sparse entry has a value with distribution of $N(0,1)$. We then plot the graph of different degree of sparsity and the corresponding effectiveness of the optimization heuristic in extracting the original pq^T. The result is as follows.