

Robust Gaussian Function Algorithm for Matrix Completion

Juan Geng¹, Mengxuan Zhao², Cuiping Nie³, Laisheng Wang^{4*},

¹*College of mathematics and statistics, Hebei University of Economics and Business,
050064 Shijiazhuang, China
hebeigengjuan@163.com*

²*Shijiazhuang Jingtong Culture Communication Limited Company,
050064 Shijiazhuang, China
sxxzmx11@126.com*

³*Department of Mathematics and Information Science, Shijiazhuang College,
050035 Shijiazhuang, China
niecuiping7@163.com*

⁴*College of Science, China Agricultural University, 100083 Beijing, China
wanglaish@126.com*

Abstract

This paper introduces a novel algorithm to solve the matrix rank minimization problem among all matrices obeying a set of convex constraints. The most popular convex relaxation of the rank minimization problem minimizes the nuclear norm instead of the rank of the matrix. In this paper we are interested in using robust Gaussian function to solve the low-rank matrix completion problem, which is the special case of the rank minimization problem. This regularized problem is a differential smooth convex optimization problem, in which the objective function is the sum of a convex smooth function. The numerical results suggest that our algorithm is efficient and robust in solving randomly generated matrix completion problems. Finally, we test our algorithm on low-rank image recovery problem.

Keywords: *Matrix completion, Gaussian function, cut error technology, image recovery*

1. Introduction

The low-rank matrix completion problem usually can be regarded as a generalization compressive sensing of vector which has been attracting a lot of researchers during the last years. It has widely applied in lots of fields, such as collaborative filtering, triangulation from incomplete data and linear system identification [1-2]. The matrix completion problem is to reconstruct an unknown matrix with low-rank or approximately low-rank constraints from a small set of observed entries. In this paper, we consider the following optimization problem:

$$\begin{aligned} \min \quad & \text{rank}(X) \\ \text{s. t.} \quad & X_{ij} = M_{ij}, \quad (i, j) \in \Omega \end{aligned} \quad (1.1)$$

where $X, M \in \mathbb{R}^{n^1 \times n^2}$ and Ω is a subset of index pairs (i, j) . Candès [1] proves that most low-rank matrix completion problems can be perfectly recovered by solving the optimization problem

* Corresponding author.

$$\begin{aligned} \min \quad & \|X\|_* \\ \text{s.t.} \quad & X_{ij} = M_{ij}, \quad (i, j) \in \Omega \end{aligned} \quad (1.2)$$

In recent years, many researchers develop a number of efficient methods to solve the above model by using the nuclear norm of the matrix [3]. Ghasemi [4] proposes a smooth function method to solve the matrix completion. They mainly consider the optimization problem:

$$\begin{aligned} \min \quad & n - \sum_{k=1}^n e^{-\sigma_k^2(X)/2\delta^2} \\ \text{s.t.} \quad & X_{ij} = M_{ij}, \quad (i, j) \in \Omega \end{aligned} \quad (1.3)$$

where $\sigma_k(X)$ is the k -th singular value of the matrix X , which is achieved by the singular value decomposition (SVD) of matrix. In this model, we can obviously see that when $\sigma_k(X) = 0$, $e^{-\sigma_k^2(X)/2\delta^2} = 1$; when $\sigma_k(X) \neq 0$ and δ is enough small, $e^{-\sigma_k^2(X)/2\delta^2}$ approaches to zero. We observe that $n - \sum_{k=1}^n e^{-\sigma_k^2(X)/2\delta^2}$ approximates to the rank of the matrix X only if δ is enough small. However, as δ decreasing it will result in a highly non-smooth about $n - \sum_{k=1}^n e^{-\sigma_k^2(X)/2\delta^2}$ with many local minima, and the gradient projection method might be captured in local minima. Then the optimization (1.3) will fail to converge. In contrast, for large values of δ , the approximation of the rank function will not be accurate enough. To overcome this problem, Ghasemi decrease δ in each iteration gradually [4]. In this paper, we add a parameter in the function $e^{-\sigma_k^2(X)/2\delta^2}$ and with the parameter, $n - \sum_{k=1}^n e^{-\sigma_k^2(X)/2\delta^2}$ has a better representation to approximate the rank function.

2. Main Algorithm

2.1. Improved Ideas

Before introducing our algorithm, two points should be noticed. The first is how to choice the parameter C ; The second is how to control the influence of the smaller singular value to the objective function. To solve the first problem, we will gradually decrease in each iteration. This technique, used for minimizing non-convex functions, is referred to as GNC (Graduated Non-Convexity) [5]. In our algorithm, we choose the biggest singular value as δ gradually decrease in each iteration, and add another parameter C in the objective function. Set $F_{C\delta}(X) = n - \sum_{k=1}^n e^{-C\sigma_k^2(X)/2\delta^2}$, then it has a better representation to approximate the rank function with the parameter C . In order to illustrate our idea more clearly, we plot the Gaussian function with parameter in Figure 1. From the figure, we can see that when the parameter C is enough big, the objective function will approximate the rank function better.

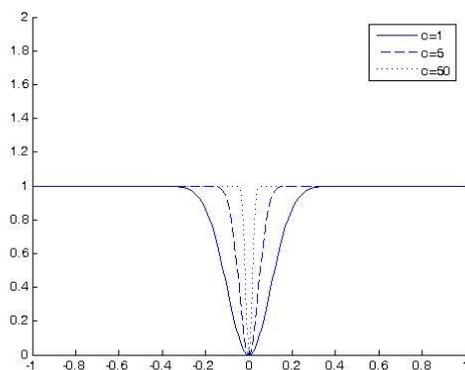


Figure 1. The Comparison of Function $1 - e^{-Cx^2/2\delta^2}$ with Different C

Now we are interested in the method for solving the following matrix completion problem:

$$\begin{aligned} \min \quad & F_{C\delta}(X) \\ \text{s. t.} \quad & X_{ij} = M_{ij}, \quad (i, j) \in \Omega \end{aligned} \quad (2.1)$$

where $F_{C\delta}(X) = n - \sum_{k=1}^n e^{-C\sigma_k^2(X)/2\delta^2}$ and Ω is a subset of index pairs (i, j) . To solve the second problem, we will assume $\sigma_i(X) = 0$ if $\sigma_i(X) < \varepsilon$, (ε is a small positive number). We will choice ε based on different C.

2.2. Robust Gaussian Function Algorithm

For simplicity, we assume that the unknown rank-r matrix M is $n \times n$ for which only m entries can be observed. According to [1] it can be reconstructed by minimizing the nuclear norm function with probability at least $1 - cn^3$ if $m > cn^{\frac{5}{4}} \log n$ (where c is a constant).

Our algorithm to solve matrix completion problem is to use the improved Gaussian function instead of the nuclear norm to approximate the rank function. Then use the gradient projection method to solve this problem. We know that $F_{C\delta}(X): \mathbb{R}^{n^1 \times n^2} \rightarrow \mathbb{R}$ is a continuous approximation of the rank function, and it is differentiable, so a gradient projection method can be used to solve the problem (2.1).

Definition 1 (Absolutely Symmetric [6]). A function $f: \mathbb{R}^q \rightarrow [-\infty, +\infty]$ is called absolutely symmetric if $f(\gamma) = f(\hat{\gamma})$ for any vector γ in \mathbb{R}^q . Where $\hat{\gamma}$ is arbitrary permutations and sign changes of the components of γ .

Corollary 1 (Characterization of Subgradients [6, Cor2.5]). Suppose the function $f: \mathbb{R}^q \rightarrow [-\infty, +\infty]$ is absolutely symmetric, and the $m \times n$ matrix X has $\sigma(X)$ in $\text{dom}f$. Then the $m \times n$ matrix Y lies in $\partial(f \circ \sigma)(X)$ if and only if $\sigma(Y)$ lies in $\partial f(\sigma(X))$ and there exists a simultaneous singular value decomposition of the form

$$X = V(\text{Diag } \sigma(X))U \quad \text{and} \quad Y = V(\text{Diag } \sigma(Y))U,$$

with U in \mathcal{U}_n and V in \mathcal{U}_m , where \mathcal{U}_n denotes the set of $n \times n$ unitary matrices. In fact

$$\partial(f \circ \sigma)(X) = \{V(\text{Diag } \mu)U \mid \mu \in \partial f(\sigma(X)), U \in \mathcal{U}_n, V \in \mathcal{U}_m, X = V(\text{Diag } \sigma(X))U\}. \quad (2.2)$$

Theorem 1. For matrix M $\in \mathbb{R}^{n^1 \times n^2}$ with singular value decomposition (SVD) $X = U\text{Diag}(\sigma_1, \sigma_2, \dots, \sigma_n)V^H$, the gradient of $F_{C\delta}(X)$ is

$$\frac{\partial F_{C\delta}(X)}{\partial X} = U \text{Diag} \left\{ \frac{C\sigma_1}{\delta^2} e^{-\frac{C\sigma_1^2}{2\delta^2}}, \frac{C\sigma_2}{\delta^2} e^{-\frac{C\sigma_2^2}{2\delta^2}}, \dots, \frac{C\sigma_n}{\delta^2} e^{-\frac{C\sigma_n^2}{2\delta^2}} \right\} V^H \quad (2.3)$$

Proof: Let $F_{C\delta}(X) = f(\sigma_C(X)) = f\sigma_C(X)$, where $\sigma_C(X): \mathbb{R}^{n^1 \times n^2} \rightarrow \mathbb{R}^n$ has the singular value of the matrix X and $f(y) = n - \sum_{k=1}^n e^{-Cy_k^2/2\delta^2}$. According to the Corollary 1, the sub-differential of $F_{C\delta}(X)$ is given by:

$$\partial F_{C\delta}(X) = \{U(\text{Diag } \mu)V^H \mid \mu \in \partial f(\sigma_C(X))\}.$$

Since $\partial f(\sigma_C(X))$ is differentiable at $\sigma_C(X)$, $\partial f(\sigma_C(X))$ is a singleton and consequently it is converted to $\nabla f(\sigma_C(X))$ and equation (2.3) is obtained.

For a given matrix X , we define the projection operator $\mathcal{A}: \mathbb{R}^{n^1 \times n^2} \rightarrow \mathbb{R}^{n^1 \times n^2}$ [7]:

$$\mathcal{A}(X) = \begin{cases} M_{ij}, & (i, j) \in \Omega \\ X_{ij}, & (i, j) \notin \Omega \end{cases} \quad (2.4)$$

Where Ω is a subset of index pairs (i, j) , and M can be regarded as a sample matrix. So the gradient projection operator at the k -th iteration will be

$$X_{k+1} = \mathcal{A}(X_k - \mu_k \frac{\partial F_{C\delta_k}(X_k)}{\partial X_k}) \quad (2.5)$$

Where μ_k is the step size at the k -th iteration, and δ_k is the biggest singular value of X_k . The gradient projection method can be shown as the following three steps by considering the gradient given in (2.5).

Step 1: Singular value decomposition of matrix X_k :

$$X_k = U_k \text{Diag}(\sigma_1, \sigma_2, \dots, \sigma_n) V_k^H$$

Step 2: Gradient projection:

$$X_{k+1} = U_k \text{Diag} \left\{ \sigma_1 - \frac{C\sigma_1}{\delta^2} e^{-\frac{C\sigma_1^2}{2\delta^2}}, \dots, \sigma_n - \frac{C\sigma_n}{\delta^2} e^{-\frac{C\sigma_n^2}{2\delta^2}} \right\} V_k^H$$

Step 3: Projection operator:

$$X_{k+1} = \mathcal{A}(X_{k+1})$$

Then we give the Pseudo code of our algorithm which we called Robust Gaussian function Algorithm (RGF).

Table 1. Robust Gaussian Function Algorithm (RGF)

Algorithm 1. Robust Gaussian function Algorithm (RGF)

Inputs: $X = M_0 = \begin{cases} M_{ij}, & (i, j) \in \Omega \\ 0, & (i, j) \notin \Omega \end{cases}$ (initialization matrix), error, $d = 0, L$

Output: X_{opt}

While $d > \text{error}$ **do**

 Let $X_0 = X$

Inner loop:

For $i = 1:L$

 Compute $X_k = U_k \text{Diag}(\sigma_1, \sigma_2, \dots, \sigma_n) V_k^H$

for $i = 1:n$

 if $\sigma_i < 0.3$, then $\sigma_i = 0$

end

$$X = U \text{Diag} \left\{ \sigma_1 - \frac{C\sigma_1}{\delta^2} e^{-\frac{C\sigma_1^2}{2\delta^2}}, \dots, \sigma_n - \frac{C\sigma_n}{\delta^2} e^{-\frac{C\sigma_n^2}{2\delta^2}} \right\} V^H.$$

 Compute projection $X = \mathcal{A}(X)$

End

$d = \|X - X_0\|_F / \|X\|_F.$

End

$X_{opt} = X.$

3. Numerical Experiment

In this part, there will be a detail introduction of the algorithm RGF and the performance of the algorithm RGF compared to SRF [4]. Our implementation is in Matlab (Version R2010b) and all the computational results we reported are obtained on a desktop computer with a 2.40GHz CPU and 2 GB of memory. In our numerical experiment, we generate $n \times n$ matrix of rank r by sampling two $n \times r$ factors M_L and M_R independently, each having i.i.d. Gaussian entries, and let $M = M_L M_R^H$ as it is suggested in [1]. The set of sample entries Ω is sampled uniformly at random among all sets of cardinality m , and we denote d_r the degree of freedom which is computed by $d_r = (2n - r)r$.

In the following experiments, we use

$$\|X_{opt} - M\|_F / \|M\|_F < 10^{-4}$$

as the stopping criterion in RGF algorithm. As discussed above, the step sizes are constant and we set $\mu_k = 1.8$. Throughout this part, we denote the output of the RGF algorithm by X_{opt} . The parameter C is chosen empirically and we set $C = 50$.

The computational results of RGF and SRF are displayed in Table 2 and Table 3. In the tables, n is the dimension of the unknown matrix M and r is the rank of the matrix M . m/d_r is the ratio between the number of sampled entries and the number of degrees of freedom. Time denotes the CPU time in second. The error is the value of $\|X_{opt} - M\|_F / \|M\|_F$. In Table 2, we solve the easy problems in which the ratio m/d_r is larger than 3. And the experimental results of the hard problems in which the ratio is less than 3 are reported in Table 3. From the tables, we obviously see that no matter the easy problem or the hard problem, our improved Gaussian function method (RGF) is faster at least 50% than SRF.

Table 2. Numerical Results on the Easy Problems

Problem (n, r, m/d _r)	RGF		SRF	
	Time (s)	Error (10 ⁻⁷)	Time (s)	Error (10 ⁻⁷)
(50,5,4)	0.14	0.79	0.14	4.6
(100,5,4)	1.16	5.2	1.12	9.8
(100,10,3.5)	0.98	3.7	0.89	8.4
(200,10,4)	8.5	8.5	8	12
(200,20,3)	7.5	3.2	7.7	7.7
(500,20,4)	450	37	507	72
(500,50,3)	124	0.90	156	2.9
(1000,50,4)	1243	3.1	1405	6.0
(1000,100,3)	958	1.9	1075	4.2

Table 3. Numerical Results on the Hard Problems

Problem (n, r, m/d _r)	RGF		SRF	
	Time (s)	Error (10 ⁻⁶)	Time (s)	Error (10 ⁻⁶)
(50,5,2)	0.7	18	0.8	26
(100,5,2.8)	3.6	18	5.4	33
(100,10,1.8)	5.8	19	3.4	247
(200,10,3.1)	10.6	1.9	17	5.7
(200,20,2.1)	13.7	1.9	20	5.9
(500,20,3.2)	1174	6.26e-2	1493	1.14e-1

(500,50,1.8)	443	0.92	611	7.41
(1000,50,3.2)	1442	0.58	2155	1.61
(1000,100,1.8)	3241	0.62	4023	3.16

4. Image Simulation

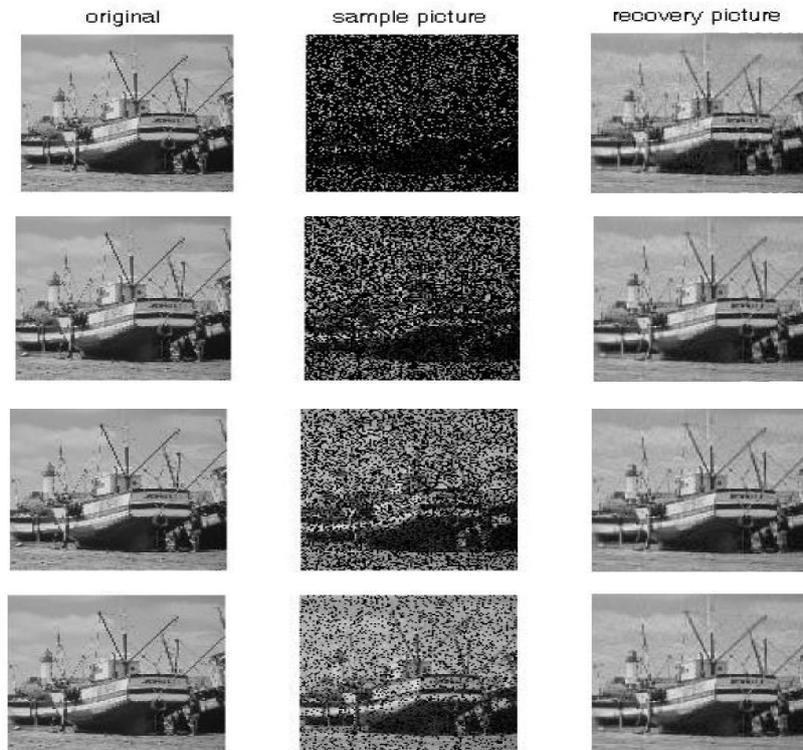


Figure 2. Low-Rank Image Recovery Using RGF

The first line shows the original image, sampled image ($p = 0.2$) and the recovered image. Similarly, the second line of the figure shows the sampled image ($p = 0.4$) and the recovered image. The third line and the fourth line are the images when $p = 0.6$ and $p = 0.8$, respectively.

As an application to image processing, we test the effect of the algorithm RGF in low-rank image recovery problem. In this part, we test the image “Boat” that has 512×512 pixels with full rank and has been widely used in many simulations [8], since the image has a nice mixture of details, shading area, flat regions and textures. In Figure 2, the value of p expresses the ratio of sampled entries in the image. For example, $p = 0.2$ expresses that we only know the 20 percent pixels of the original picture. Figure 2 shows the recovered performance of the algorithm RGF in different situations when $p = 0.2, 0.4, 0.6, 0.8$. We see the image entries as a matrix, and sample p pixels of the image uniformly at random. Then we regard the sample pixels as a sample matrix. We can reconstruct the sample matrix (sample image) [9] by the RGF algorithm. From the figure, it can be seen that our algorithm can reconstruct the sample image even the sample ratio has only 20%.

5. Conclusion

In this paper, we propose an improved algorithm for solving the matrix completion problem. In our algorithm, we improved two points based on the Gaussian function. First,

we add a parameter on the Gaussian function to improve the accuracy. Second, how to choice δ in the algorithm SRF is a discussed. Decreasing δ to zero will results in a highly non-smooth Gaussian function with many local minima. But for the large value of δ , the approximation of the rank function will not be accurate enough. To solve this problem we add the parameter C in the Gaussian function. In the algorithm, we use cut error technology to avoid the influence of the small singular value. The proposed algorithm has a better performance in the numerical experimentation and the image simulation than the previous algorithm SRF.

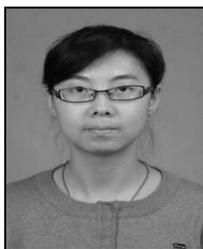
Acknowledgements

The authors would like to thank the anonymous reviewers for the useful comments and suggestions. This work is partially supported by National Natural Science Foundation of China Grant (No. 11271367), the Scientific Research Foundation of Hebei Education Department (No. QN2015052), Science and Technology Project of Hebei Province (No. 15215327), Science and Technology Research Foundation of Hebei University of Economics and Business (No. 2015KYZ05) and Natural Science Foundation of Hebei Province (No. A2015106040).

References

- [1] E. J. Candès and B. Recht, "Exact Matrix Completion via Convex Optimization. Foundations of Computational Mathematics", vol. 9, (2009), pp. 717-772.
- [2] B. Recht, M. Fazel and P. Parrilo, "Guaranteed minimum rank solutions of matrix equations via nuclear norm minimization", SIAM Rev, vol. 52, no. 3, (2010), pp. 471-501.
- [3] J. F. Cai, E. J. Candès and Z. W. Shen, "A singular value thresholding algorithm for matrix completion", SIAM J. Optim., vol. 20, (2010), pp. 1956-1982.
- [4] H. Ghasemi, M. Babaie-Zadeh and C. Jutten, "SRF: Matrix completion based on smoothed rank function, in Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on (2011).
- [5] R. H. Keshavan, A. Montanari and Oh. Sewoong, "Matrix Completion from a Few Entries", IEEE Trans. On Information Theory, vol. 56, no. 1, (2010), pp. 2980-2998.
- [6] A. S. Lewis, "The convex analysis of unitarily invariant matrix norms", Journal of Convex Analysis, vol. 2, (1995), pp. 173-183.
- [7] E. J. Candès and T. Tao, "The power of convex relaxation: near-optimal matrix completion", IEEE Trans. on Information Theory, vol. 56, no. 5, (2010), pp. 2053-2080.
- [8] S. Q. Ma, D. Goldfarb and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization", Mathematical Programming, vol. 128, no. 1, (2011), pp. 321-353.
- [9] Z. W. Wen, W. T. Yin and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm", Mathematical Programming Computation, vol. 4, no.4, (2012), pp. 333-361.

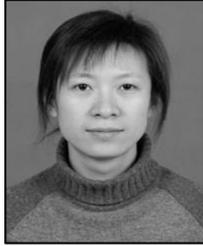
Authors



Geng Juan. She received her B. S. degree (2003) and M. S. degree (2006) from Hebei Normal University, China. In 2014, she received the Ph.D. degree from China Agriculture University. She is currently teaching in College of mathematics and statistics at Hebei University of Economics and Business. Her research interests include matrix rank minimization, tensor completion and data mining.



Zhao Mengxuan. She received her B. S. degree from Xingtai University, China (2011) and M. S. degree from China Agriculture University (2013). Her research interests include matrix rank minimization, tensor completion and image processing.



Nie Cuiping. She received her B. S. degree (2003) and M. S. degree (2006) from Hebei Normal University, China. In 2012, she received the Ph.D. degree from School of Management and Economics at Beijing Institute of Technology. She is currently teaching in Department of Mathematics and Information Science at Shijiazhuang College. Her research interests include fuzzy cooperation games.



Wang Laisheng. He received the B. S. degree from Jilin University, China (1982) and M. S. degree from Jilin University, China (1989). He received the Ph.D. degree from China Agricultural University, Beijing, China, in 2001. Currently, he is a professor at the school of science, China Agricultural University, Beijing, China. His research interests include data mining, image processing and pattern recognition.