Supplementary Material for Discriminative Regression with Adaptive Graph Diffusion

Jie Wen, Shijie Deng, Lunke Fei, Zheng Zhang, *Senior Member, IEEE*, Bob Zhang, *Senior Member, IEEE*, Zhao Zhang, *Senior Member, IEEE*, Yong Xu, *Senior Member, IEEE*

I. Running time analysis

Table I lists the running times (seconds) of all compared methods and the proposed DRAGD on the six databases mentioned in the main document, where each class has 20, 20, 12, 20, 20, and 7 training samples. For a fair comparison, all methods are implemented on the same computer with MATLAB 2015a, 64 GB RAM, Intel Core i9-9900k CPU, and a Wondows 10 operating system. From the table, we can observe that the computation efficiency of our method relies on the feature dimension of the input samples, while the computational efficiencies of the other methods except the representation-based classification methods are highly affected by the number of training samples. According to the practical running times listed in the table, our method performs faster than many related methods, such as SLRR, MSRL, CLSR, LRC, and SRC, where MSRL and CLSR are the two competitive advanced linear regression-based classification methods. Although the proposed method performs slightly slower than DLSR, LRLR, and LRRR, the classification accuracy obtained by our DRAGD is much higher than those of the three methods, as shown in Tables II-VII in the main manuscript.

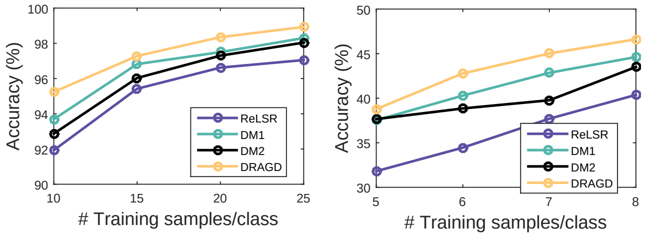
II. Ablation study

From objective model (#4)[1], the proposed DRAGD mainly introduces two schemes, adaptive graph embedding and retargeted learning, to enhance the classification performance of linear regression. In this section, we compare DRAGD with ReLSR and two derived degradation models of our DRAGD, called DM1 and DM2, on the COIL20 and LFW databases, to validate the effectiveness of the two exploited techniques. The models of DM1 and DM2 are expressed as follows:

$$DM1: \begin{array}{l} \min\limits_{T,W} \left\| T - WX \right\|_F^2 + \frac{\lambda_2}{2} \left\| W \right\|_F^2 \\ + \frac{\lambda_1}{2} \sum\limits_{i,j=1}^{n} \left\| Wx_i - Wx_j \right\|_2^2 A_{i,j} \\ s.t. \forall \left\{ i, j, j \neq l_i \right\}, T_{l_i.i} - T_{j,i} \geq 1 \end{array} \tag{1}$$

$$DM2: \begin{array}{l} \min\limits_{W,A} \left\| Y - WX \right\|_F^2 + \frac{\lambda_1}{2} \sum\limits_{i,j=1}^{n} \left\| Wx_i - Wx_j \right\|_2^2 A_{i,j} \\ + \frac{\lambda_2}{4} \sum\limits_{p,q=1}^{n} \sum\limits_{i,j=1}^{n} Z_{i,j} Z_{p,q} \left(\frac{A_{i,p}}{\sqrt{D_{i,i} D_{p,p}}} - \frac{A_{j,q}}{\sqrt{D_{j,j} D_{q,q}}} \right) \\ + \frac{\lambda_3}{2} \left\| W \right\|_F^2 \\ s.t. 0 \leq A \leq 1, diag\left(A \right) = 0, A^T = A, A\mathit{1} = \mathit{1} \end{array} \tag{2}$$

For DM1, we mainly utilize a conventional graph embedding constraint with a preconstructed nearest neighbor graph A from the training data to replace the adaptive graph learning term of our DRAGD. For DM2, we simply remove the retargeted learning scheme by using a fixed one-hot label matrix Y as the conventional linear regression. The experimental results of DRAGD, DM1, and DM2 on the COIL20 and LFW datasets are plotted in Fig.1. It is obvious that: 1) DRAGD outperforms DM1 and DM2 on the two datasets. 2) DM1 performs better than ReLSR. These experimental results demonstrate that: 1) Graph embedding and retargeted learning are two effective schemes to enhance the performance of linear regression-based classification tasks. 2) Our proposed graph learning and embedding term can learn a more informative graph than the conventional nearest neighbor-based graph construction approach.



(a) COIL20

(b) LFW

Fig. 1. Experimental results of ReLSR, DRAGD and its two degradation methods on the COIL20 and LFW databases.

III. The proof of proposition 1

Firstly, it is obvious that the second term $\sum\limits_{i,j=1}^{n} \left\| Wx_i - Wx_j \right\|^2 A_{i,j}$ explores the local distance structure

This work is partially supported by National Natural Science Foundation of China under Grant 62006059 and 62176066, Shenzhen Fundamental Research Fund under Grant JCYJ20190806142416685, and Shenzhen Science and Technology Program under Grant RCBS20210609103709020.

Jie Wen, Shijie Deng, Zheng Zhang, and Yong Xu are with the Shenzhen Key Laboratory of Visual Object Detection and Recognition, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China. (Email: jiewen_pr@126.com; 21S151151@stu.hit.edu.cn, darrenzz219@gmail.com; yongxu@ymail.com).

Lunke Fei is with the School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China (Email: flksxm@126.com).

Bob Zhang is with the PAMI Research Group, Department of Computer and Information Science, University of Macau, Taipa, Macau, China (Email: bobzhang@um.edu.mo).

Zhao Zhang is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230000, China. (Email: cszzhang@gmail.com).

[1] '(#i)' denotes the equation (i) in the main document

TABLE I

Running time (seconds) of different regression-based methods on the COIL20, COIL100, AR, YaleB, PIE, and LFW databases with 20, 20, 12, 20, 20, and 7 training samples per class, respectively. N denotes the total number of training samples and test samples. n_i denotes the number of training samples of the ith class. 'No.' denotes the number of training samples per class.

Database (No.)	COIL20 (20)	COIL100 (20)	AR (12)	YaleB (20)	PIE (20)	LFW (7)	Computational complexity
LRLR [1]	0.797	0.893	6.524	0.903	0.888	0.928	$O\left(mn+mn^2\right)$ [2]
LRRR [1]	0.631	0.756	4.642	0.733	0.760	0.730	$O\left(mn+mn^2\right)$ [2]
SLRR [1]	72.924	27.556	107.227	72.234	15.703	41.874	$O\left(mn+mn^2\right)$ [2]
DLSR [3]	0.284	2.256	2.647	0.609	1.699	0.523	$O\left(\left(n+c\right)\left(m+1\right)^2+2mnc+nc\right)$ [3]
ReLSR [4]	0.596	12.780	12.718	1.497	5.223	2.844	$O\left(2m^2n+m+\tau\left(2mnc+nc\right)\right)$ [5]
MSRL [2]	1.834	29.844	30.784	5.820	14.444	6.943	$O\left(\tau\left(mnc+mn^2+m^2n\right)\right)$ [2]
GReLSR [6]	0.757	5.792	5.890	1.573	3.308	2.044	$O\left(c\log c\right)$ [6]
CLSR [5]	4.209	56.527	53.748	9.539	25.126	14.799	$O\left(2m^2n+m+\tau\left(2mnc+nc+n\right)\right)$ [5]
LRC [7]	2.276	59.268	26.385	7.090	79.496	3.734	$O\left(\left(N-n\right)\sum\limits_{i=1}^{c}n_i^3\right)$
CRC [8]	0.531	16.841	7.389	1.807	20.979	0.862	$O\left(\left(N-n\right)n^3\right)$
SRC [9]	24.625	22.953	11.734	58.321	29.255	1.034	$O\left(\left(N-n\right)\left(n^2+mn\right)\right)$ [2]
SVM [10]	0.778	20.831	13.823	2.569	20.122	1.291	$O\left(\tau mn\right)$ [10]
DRAGD	0.838	20.627	7.174	2.873	9.211	3.310	$O\left(\tau m^3\right)$

of data in the target subspace. Here, we focus on the proof of that our method can capture the representation structure. Defining $WX=T$, then the second term can be transformed into $\sum\limits_{i,j=1}^{n}\left\|Wx_i-Wx_j\right\|^2A_{i,j}=\sum\limits_{i,j=1}^{n}\left\|t_i-t_j\right\|^2A_{i,j}$, where t_i and t_j denote the ith and jth column vectors of T, respectively. Then, we have:

$$\begin{aligned}&\sum_{i,j=1}^{n}\left\|t_i-t_j\right\|^2A_{i,j}\\=&\sum_{i=1}^{n}\left(\sum_{j=1}^{n}A_{i,j}+\sum_{j=1}^{n}A_{j,i}\right)t_i^Tt_i-2\sum_{i,j=1}^{n}A_{i,j}t_i^Tt_j\end{aligned}\tag{3}$$

According to the boundary constraint $A\mathit{1}=\mathit{1}$ and $A^T=A$, we have $\sum\limits_{j=1}^{n}A_{i,j}+\sum\limits_{j=1}^{n}A_{j,i}=2$, then:

$$\begin{aligned}&\sum_{i,j=1}^{n}\left\|t_i-t_j\right\|^2A_{i,j}\\=&\sum_{i=1}^{n}2t_i^Tt_i-2\sum_{i,j=1}^{n}A_{i,j}t_i^Tt_j\\=&2\sum_{i=1}^{n}\left(t_i^Tt_i-\sum_{j=1}^{n}A_{i,j}t_i^Tt_j\right)\\=&2\sum_{i=1}^{n}t_i^T\left(t_i-\sum_{j=1}^{n}A_{i,j}t_j\right)\\=&2\sum_{i=1}^{n}t_i^T\left(t_i-TA_{:,i}\right)\end{aligned}\tag{4}$$

From (4), we can obtain that minimizing the graph constraint term $\sum\limits_{i,j=1}^{n}\left\|Wx_i-Wx_j\right\|^2A_{i,j}$ is equivalent to minimizing $2\sum\limits_{i=1}^{n}t_i^T\left(t_i-TA_{:,i}\right)$. Moreover, it is obvious that when $t_i=\sum\limits_{j=1}^{n}A_{i,j}t_j=TA_{:,i}$, $2\sum\limits_{i=1}^{n}t_i^T\left(t_i-TA_{:,i}\right)$ will obtain the minimum value 0. As we all known, $t_i=\sum\limits_{j=1}^{n}A_{i,j}t_j=TA_{:,i}$ is a well-known data representation/reconstruction formula, where $A_{i,j}$ denotes the reconstruction/representation coefficient of data point t_j in the collaborative representation for data point t_i. Thus we can conclude that minimizing the second graph embedding term with the boundary constraint $\left\{0\leq A\leq 1, diag\left(A\right)=0, A^T=A, A\mathit{1}=\mathit{1}\right\}$, our method can simultaneously capture the local distance structure and representation structure of data. Thus, the proposition 1 is proved.

IV. The proof of proposition 2

For the second term in (#11), we can transform it as follows:

$$\begin{aligned}&\sum_{p,q=1}^{n}\sum_{i,j=1}^{n}Z_{i,j}Z_{p,q}\left(\frac{S_{i,p}}{\sqrt{D_{i,i}D_{p,p}}}-\frac{S_{j,q}}{\sqrt{D_{j,j}D_{q,q}}}\right)^2\\=&\sum_{\alpha,\beta=1}^{n^2}\mathcal{M}_{\alpha,\beta}\left(\frac{vec(S)_\alpha}{\sqrt{\mathcal{D}_{\alpha,\alpha}}}-\frac{vec(S)_\beta}{\sqrt{\mathcal{D}_{\beta,\beta}}}\right)^2\\=&2\sum_{\alpha,\beta=1}^{n^2}\mathcal{M}_{\alpha,\beta}\frac{\left(vec(S)_\alpha\right)^2}{\mathcal{D}_{\alpha,\alpha}}\\&-2\sum_{\alpha,\beta=1}^{n^2}vec(S)_\alpha\frac{\mathcal{M}_{\alpha,\beta}}{\sqrt{\mathcal{D}_{\alpha,\alpha}}\sqrt{\mathcal{D}_{\beta,\beta}}}vec(S)_\beta\\=&2\sum_{\alpha,\beta=1}^{n^2}\left(vec(S)_\alpha\right)^2-2vec(S)^T\mathcal{D}^{-1/2}\mathcal{M}\mathcal{D}^{-1/2}vec\left(S\right)\\=&2vec(S)^T\left(I-\mathcal{D}^{-1/2}\mathcal{M}\mathcal{D}^{-1/2}\right)vec\left(S\right)\end{aligned}\tag{5}$$

where $\mathcal{M}\in R^{n^2\times n^2}=Z\otimes Z$, '$\otimes$' denotes the Kronecker product operation, $\mathcal{D}\in R^{n^2\times n^2}=D\otimes D$.

According to (5), we can obtain that

$$\begin{aligned}&\min_{S}\frac{\mu}{2}\left\|A-S+\frac{C}{\mu}\right\|_F^2\\&+\frac{\lambda_2}{4}\sum_{p,q}^{n}\sum_{i,j}^{n}Z_{i,j}Z_{p,q}\left(\frac{S_{i,p}}{\sqrt{D_{i,i}D_{p,p}}}-\frac{S_{j,q}}{\sqrt{D_{j,j}D_{q,q}}}\right)^2\\&\Leftrightarrow\min_{S}\frac{\mu}{2}\left\|vec\left(S\right)-vec\left(A+\frac{C}{\mu}\right)\right\|_F^2\\&+\frac{\lambda_2}{2}vec(S)^T\left(I-\mathcal{D}^{-0.5}\mathcal{M}\mathcal{D}^{-0.5}\right)vec\left(S\right)\end{aligned}\tag{6}$$

Thus, we complete the proof.

V. The proof of proposition 3

Before proving proposition 3, we can simply obtain that:

$$\begin{aligned}&\bar{\mathcal{Z}}_{\alpha,\beta}=\bar{Z}_{i,j}\bar{Z}_{p,q}\\&=\left(D_Z\right)_{i,i}^{-0.5}Z_{i,j}\left(D_Z\right)_{j,j}^{-0.5}\left(D_Z\right)_{p,p}^{-0.5}Z_{p,q}\left(D_Z\right)_{q,q}^{-0.5}\\&=\mathcal{D}_{\alpha,\alpha}^{-1/2}\mathcal{M}_{\alpha,\beta}\mathcal{D}_{\alpha,\alpha}^{-1/2}\end{aligned}\tag{7}$$

And thus, we have $\bar{\mathcal{Z}}=\bar{Z}\otimes\bar{Z}=\mathcal{D}^{-1/2}\mathcal{M}\mathcal{D}^{-1/2}$.

First, transforming the matrix into vector at the both sides of (#16), then we have:

$$\begin{aligned}&vec\left(S^{(t+1)}\right)=\gamma vec\left(\bar{Z}S^{(t)}\bar{Z}^T\right)+\left(1-\gamma\right)vec\left(A+C/\mu\right)\\&=\gamma\left(\bar{Z}\otimes\bar{Z}\right)vec\left(S^{(t)}\right)+\left(1-\gamma\right)vec\left(A+C/\mu\right)\\&=\gamma\bar{\mathcal{Z}}vec\left(S^{(t)}\right)+\left(1-\gamma\right)vec\left(A+C/\mu\right)\\&=\left(\gamma\bar{\mathcal{Z}}\right)^t vec\left(S^{(1)}\right)+\left(1-\gamma\right)\sum_{i=0}^{t-1}\left(\gamma\bar{\mathcal{Z}}\right)^i vec\left(A+C/\mu\right)\end{aligned}\tag{8}$$

For the normalized matrix $\bar{Z}$, it is obvious that its spectral radius is no larger than 1. In addition, based on the following **Lemma 1**, we can obtain that the eigenvalues of the Kronecker product of $\bar{\mathcal{Z}}=\bar{Z}\otimes\bar{Z}$ are in the domain of [-1,1].

Lemma 1 [11]: Let $\lambda_i\ (i=1,\dots n)$ and $\beta_j\ (j=1,\dots n)$ be the ith eigenvalue of $A\in R^{n\times n}$ and jth eigenvalue of $B\in R^{n\times n}$, respectively. Then, $\lambda_i\beta_j$ is one of the eigenvalues of the Kronecker product $A\otimes B$.

Then considering $\gamma=\frac{\lambda_2}{\mu+\lambda_2}<1$, we have:

$$\lim_{t\to\infty}\left(\gamma\bar{\mathcal{Z}}\right)^t vec\left(S^{(1)}\right)=0\tag{9}$$

$$\lim_{t\to\infty}\sum_{i=0}^{t-1}\left(\gamma\bar{\mathcal{Z}}\right)^i vec\left(A+C/\mu\right)=\left(1-\gamma\bar{\mathcal{Z}}\right)^{-1}vec\left(A+C/\mu\right)\tag{10}$$

Therefore, Eq.(8) can be further transformed as follows:

$$\begin{aligned}&\lim_{t\to\infty}vec\left(S^{(t+1)}\right)=\left(1-\gamma\right)\left(1-\gamma\bar{\mathcal{Z}}\right)^{-1}vec\left(A+C_2/\mu\right)\\&\Rightarrow\lim_{t\to\infty}S^{(t+1)}=\left(1-\gamma\right)vec^{-1}\left(\left(1-\gamma\bar{\mathcal{Z}}\right)^{-1}vec\left(A+C/\mu\right)\right)\end{aligned}\tag{11}$$

The right side of Eq.(11) is exactly the same as the right side of Eq.(#15). And thus we complete the proof.

VI. Theoretical convergence analysis

In the main document, for the objective function (#4), we adopt the well-known ALM to iteratively calculate the optimal solutions *w.r.t.* variables W, A, T, and the auxiliary S. For the objective problem with four variables, it is difficult to prove the strict convergence for the ALM-based optimization process [12, 13]. Fortunately, the following proposition 4 can illustrate a weak convergence property to our presented optimization algorithm [14, 15].

Proposition 4: The ALM-based optimization algorithm 1 for objective problem (#4) is equivalent to a two-block ALM algorithm.

Proof: A typical two-block problem solved by ALM can be formulated as follows:

$$\begin{aligned}&\min_{X,Y}f\left(X\right)+g\left(Y\right)\\&s.t.\ AX+BY=C,\\&\quad X\in\Omega\left(X\right),\\&\quad Y\in\Omega\left(Y\right)\end{aligned}\tag{12}$$

where X and Y are the variables to calculate. $f\left(\cdot\right)$ and $g\left(\cdot\right)$ are the given convex functions, $\{A,B,C\}$ are the given constants. $\Omega\left(X\right)$ and $\Omega\left(Y\right)$ are regarded as the boundary of the two variables.

For problem (12), according to ALM, the following augmented Lagrangian formula is first constructed:

$$L\left(X,Y,E\right)=f\left(X\right)+g\left(Y\right)+\frac{\mu}{2}\left\|AX-BY-C+\frac{E}{\mu}\right\|_F^2\tag{13}$$

where E and μ are the Lagrange multiplier and penalty parameter as the augmented Lagrangian formula (#5) in our main document.

Then, the optimal solution is obtained by iteratively solving two problems *w.r.t.* variables X and Y, and updating Lagrange multiplier as follows:

$$X^{t+1}=\arg\min_{X\in\Omega(X)}L\left(X,Y^t,E^t\right)\tag{14}$$

$$Y^{t+1}=\arg\min_{X\in\Omega(Y)}L\left(X^{t+1},Y,E^t\right)\tag{15}$$

$$E^{t+1}=E^t+\mu\left(AX^{t+1}-BY^{t+1}-C\right)\tag{16}$$

where $\left\{X^{t+1},Y^{t+1},E^{t+1}\right\}$ and $\left\{X^t,Y^t,E^t\right\}$ denote the value of these variables at the $t+1$th iteration and tth iteration, respectively.

In algorithm 1 of the main document, we have divided the objective problem into four major subproblems *w.r.t.* variables W, A, T, and the auxiliary S. It is easy to observe that optimizing variable S is independent with variables W and T. Moreover, optimizing problem of T only relies on the value of variable W. In this case, we can unify the optimization processes associated with variables $\{W,T,S\}$ into the optimization process of Y as in (15). Obviously, the optimization process of variable A and the iteration formula (#21) for Lagrange multiplier are similar to the optimization for variable X as in (14) and iteration formula (16). Therefore, our presented optimization algorithm 1 can be regarded as a two-block ALM optimization process.

The convergence property of ALM algorithm for the classical two-block optimization problem has been strictly proven in the previous works [12, 16-19]. Therefore, as a similar two-block optimization problem, the adopted ALM optimization algorithm 1 can converge to the local optimal solution for objective problem (#4).

References

[1] X. Cai, C. Ding, F. Nie, and H. Huang, "On the equivalent of low-rank linear regressions and linear discriminant analysis based regressions," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 1124–1132.

[2] Z. Zhang, L. Shao, Y. Xu, L. Liu, and J. Yang, "Marginal representation learning with graph structure self-adaptation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4645–4659, 2017.

[3] S. Xiang, F. Nie, G. Meng, C. Pan, and C. Zhang, "Discriminative least squares regression for multiclass classification and feature selection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 11, pp. 1738–1754, 2012.

[4] X.-Y. Zhang, L. Wang, S. Xiang, and C.-L. Liu, "Retargeted least squares regression algorithm," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 2206–2213, 2014.

[5] H. Yuan, J. Zheng, L. L. Lai, and Y. Y. Tang, "A constrained least squares regression model," *Information Sciences*, vol. 429, pp. 247–259, 2018.

[6] L. Wang and C. Pan, "Groupwise retargeted least-squares regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 1352–1358, 2017.

[7] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106–2112, 2010.

[8] L. Zhang, M. Yang, X. Feng, Y. Ma, and D. Zhang, "Collaborative representation based classification for face recognition," *arXiv preprint arXiv:1204.2358*, 2012.

[9] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2008.

[10] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.

[11] S. Bai, X. Bai, Q. Tian, and L. J. Latecki, "Regularized diffusion process for visual retrieval," in *AAAI Conference on Artificial Intelligence*, 2017, pp. 3967–3973.

[12] Y. Zhang, "Recent advances in alternating direction methods: Practice and theory," in *IPAM Workshop on Continuous Optimization*, vol. 385, 2010.

[13] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2012.

[14] Z. Zhang, Z. Lai, Y. Xu, L. Shao, J. Wu, and G.-S. Xie, "Discriminative elastic-net regularized linear regression," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1466–1481, 2017.

[15] J. Wen, Z. Zhong, Z. Zhang, L. Fei, Z. Lai, and R. Chen, "Adaptive locality preserving regression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 75–88, 2020.

[16] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*. SIAM, 1989.

[17] E. Esser, "Applications of lagrangian-based alternating direction methods and connections to split bregman," *CAM report*, vol. 9, p. 31, 2009.

[18] J. Eckstein and D. P. Bertsekas, "On the douglasłrachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1, pp. 293–318, 1992.

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