To be decided

First A. Author, *Fellow, IEEE*, Second B. Author, and Third C. Author, Jr., *Member, IEEE*

***Abstract*—Matrix Completion by nuclear norm minimization is utilized in a variety of applications ranging from collaborative filtering to data recovery in Internet of Things sensors. However, being an optimization method, it can take many iterations to converge, with each iteration involving computationally expensive singular value decomposition operation. In this work, we unfold the Augmented Lagrange Multiplier method for matrix completion introducing convolution layers in the process. Our experiments on a real-world sensing dataset show that ConvMC-Net is significantly faster in both training and at inference, and easily applicable on high-dimensional matrices while giving the same accuracy as the existing state of the art deep unfolded method for matrix completion.**

***Index Terms*—Matrix completion, data-retrieval, deep unfolding, Internet of Things, inverse problems**

I. Introduction

Low-rank property of signals is ubiquitous in real-world applications. For instance, time-series data such as videos [1] and spatio-temporal sensing data such as that from wireless sensor networks [2] are usually low-rank. Matrix completion (MC) involves recovering an incomplete low-rank matrix from a small subset of its elements. Recht *et al.* [3] proposed nuclear norm minimization (NNM), which in turn is convex relaxation of the rank minimization problem, for solving MC problem. Furthermore, Tao *et al.* [4] theoretically proved that under certain incoherence assumptions on the singular vectors of the matrix, D can be accurately recovered by NNM. In fact, different variants of NNM for MC exist depending upon the applications, ranging from sensor localization [5] to collaborative filtering on high-dimensional matrices [6]. This model-based technique imposes low-rank assumption on the underlying matrix, which in most cases reflect the actual behaviour but it can also surmount to enforcing a model which does not fully capture the underlying statistics resulting in degraded performance. Moreover, NNM being an optimization method can take hundreds of iterations to converge with each iteration involving computationally expensive singular value decomposition (SVD) operation limiting its use for real time and online applications running on hardware constrained resources deployed in Internet of Things (IoT) such as in mobile systems, unmanned aerial vehicles and wireless sensor networks.

Data driven deep neural networks (DNNs) have shown to be much more computationally efficient at inference than optimization based subspace methods. Some examples of DNNs solving Matrix Completion problems should come here. However, DNNs often require large training datasets, extreme parameterization on top of high computational burden of training [7]. This limitation again becomes particularly relevant when operating on hardware-constrained devices as such systems are typically limited in their ability to utilize highly parameterized DNNs. In addition, DNNs learn their decision mapping solely from data giving rise to generalization issues on unseen data. Consequently, DNNs do not offer the flexibility and reliability in their design, which IoT systems need to adapt to variations in the environment. The recently proposed deep unfolded neural networks, designed by mapping the iterative steps of an optimization algorithm into DNN layers with learnable parameters, aim at solving the above-mentioned issues of DNNs while avoiding the high computational cost that is associated with optimization methods taking hundreds of iterations to converge. Building upon this approach, Eldar *et al.* considered the spatial and temporal correlation of incomplete sensing data in their unfolded version of alternating direction method of multipliers (ADMM) algorithm for MC named ADMM-Net [8]. The parameters of ADMM-Net are learned from data. It used significantly lesser number of iterations than its iterative counterpart with slight accuracy loss. However, for an input of size $M \times N$, ADMM-Net requires calculating inverses of size $MN \times MN$ at every layer/iteration which still introduces considerable computational burden at both training and at inference.

In this paper, we propose the deep unfolded network of ConvMC-Net for MC. As the name suggests, ConvMC-Net uses learnable convolution kernels, which are learned from data, in its unfolding process. Our experiments on a real world dataset of a sensor network show that the proposed ConvMC-Net is significantly faster than ADMM-Net in both training and at inference while giving better accuracy and showing improved robustness. Moreover, the number of layers of ConvMC-Net can be increased very easily for improved convergence, if needed, and it is also easily applicable on high-dimensional matrices.

II. Problem Formulation

A. Matrix Completion by Augmented Lagrange Multiplier (ALM)

MC aims at restoring the original matrix $D^{M\times N}$ from its undersampled/incomplete version D_Ω, expressed as

$$D_\Omega = U \circ D, \tag{1}$$
$$U(i,j) = \begin{cases} 1 & \text{if } (i,j) \in \Omega \\ 0 & \text{otherwise} \end{cases},$$

where Ω is a subset containing 2D coordinates of sampled entries, $\circ$ represents the Hadamard product and U represents the sampling matrix. Furthermore, it is assumed that D is low-rank. Recht *et al.* [3] proposed NNM to solve MC problem as

$$\min_L \|L\|_*, \text{ such that } D_\Omega = L_\Omega, \tag{2}$$

where $\|.\|_*$ represents the nuclear norm of a matrix defined as the sum of the singular values of L and L_Ω denotes the projection on Ω. We formulate (1) in an augmented Lagrangian form as

$$\mathscr{L}(L,Y) = \|L\|_* + \mathrm{Tr}(Y^T \cdot (D_\Omega - L_\Omega)) + \frac{\mu}{2}\|D_\Omega - L_\Omega\|_F^2, \tag{3}$$

where $\|.\|_F$ denotes the Frobenius norm, $\mathrm{Tr}()$ denotes the trace of a matrix, Y is the Lagrange multiplier of size $M \times N$, and μ is a regularization parameter that controls the penalty for violation of linear constraints. The augmented Lagrangian method seeks a saddle point of $\mathscr{L}$ by alternating between minimizing with respect to the primal variable L and taking one step of gradient ascent to increase $\mathscr{L}$ using dual variable Y as

$$L^{k+1} \in \min_L \frac{1}{\mu}\|L\|_* + \frac{1}{2}\|D_\Omega - L_\Omega + \frac{Y}{\mu}\|_F^2, \tag{4}$$

$$Y^{k+1} = Y^k + \mu \nabla_Y \left(\mathscr{L}\left(L^{k+1}, Y\right)\right), \tag{5}$$

where $\nabla_Y \mathscr{L}\left(L^{k+1}, Y\right)$ denotes the gradient of (3) with respect to Y. We then solve (4) using ISTA [9], the iterative step of which is given as

$$L^{k+1} = \mathrm{prox}\left(L^k - \frac{1}{\mu}\nabla\left(L^k\right)\right), \tag{6}$$

where prox (.) denotes the proximal operator [10] and $\nabla(L)$ denotes the gradient of the quadratic part with respect to L. The proximal operator for computing the low-rank component is the singular value thresholding operation [11], $\Psi_\alpha(X) = \mathbf{U}\mathrm{diag}(\mathrm{Relu}(\sigma_i - \alpha))\mathbf{V}^H$, where X has a SVD given by $X = U\Sigma V^T$, diag(y_i) represents a diagonal matrix with its i^{th} diagonal entry equal to y_i and σ_i represents the i^{th} singular value of X. The resulting procedure is stated in Algorithm 1.

Algorithm 1: Matrix Completion by ALM

1: *Initialize*: $L^0 = Y^0 = 0$, $\mu > 0$
2: **while** not converged **do**
3: $L^{k+1} = \Psi_{\mu^{-1}}\{L_{\Omega^C} + D_\Omega + \mu^{-1}Y_\Omega\}$, where Ω^C denotes the compliment of set Ω
4: $Y^{k+1} = Y^k + \mu\left(D_\Omega - L_\Omega^{k+1}\right)$
5: **end while**
6: **return** L^{k+1}

B. The Proposed ConvMC-Net

Although there are different ways [12], [13] and [14] to unfold an iterative algorithm into a DNN, we follow the method of [12] in unfolding Algorithm 1. Hence, we introduce a measurement matrix H into (4) as follows

$$\min_L \frac{1}{\mu}\|L\|_* + \frac{1}{2}\|D_\Omega - (HL)_\Omega + \frac{Y_\Omega}{\mu}\|_F^2, \tag{7}$$

We further replace $(HL)_\Omega$ with GL_Ω on the basis that there exists a matrix G, corresponding to H, such that $\|(HL)_\Omega - GL_\Omega\|_F^2 < \epsilon$. Using (6), the iterative step for L, after some algebraic manipulation, turn out to be

$$L^{k+1} = \Psi_{\frac{1}{\mu \cdot c}}\left\{L^k + \left(G^T\right)D_\Omega + \left(-G^TG\right)L_\Omega^k + \left(\frac{G^T}{\mu^k}\right)Y_\Omega^k\right\}, \tag{8}$$

We unroll (8) into the multi-layer neural network of ConvMC-Net by replacing the functions of the matrix G with learnable parameters. Consequently, we replace G^T and G^TG, multiplied with D_Ω and L_Ω^k ,respectively with convolution kernels. In case of the Lagrange multiplier Y, we consider for a fixed G, two matrices W and B such that $G^TY_\Omega = W \circ Y_\Omega + B$. Finally, we write the deep unfolded version of (7) as

$$L^{k+1} = \Psi_{\frac{1}{\mu^k}}\left\{L^k + \left(C_1^k * D\right) + \left(C_2^k * L_\Omega^k\right) + \left(W^k \circ Y_\Omega^k + B^k\right)\right\} \tag{9}$$

where $*$ denotes the convolution operation, C_1^k, C_2^k are the convolution kernels learned in the k^{th} layer of ConvMC-Net together with the matrices W^k and B^k and the regularization parameter $1/\mu^k$. For all our experiments, we train ConvMC-Net for 40 epochs using ADAM optimizer with a learning rate 0.01. Furthermore, we choose kernels of size (3,3) with stride (1,1), padding (1,1) and a bias. The Lagrange multiplier Y is updated in the same way as done in step 4 of Algorithm 1. We choose average normalized mean square error (NMSE) for the loss function $\mathcal{L}$ as:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\frac{\|L_i - \overline{L_i}\|}{\|L_i\|_F^2}, \tag{10}$$

where N is the total number of training sequences in the dataset, $\overline{L}$ is output predicted by the network and L is the ground-truth.

C. ADMM-Net

ADMM-Net is derived from unfolding the iterative steps for solving the following minimization problem:

$$\min_{L,Z} \rho\|Z\|_* + \frac{1}{2}\|D_\Omega - L_\Omega\|_F^2 + \frac{\lambda_1}{2}\|LT\|_F^2 + \frac{\lambda_2}{2}\|SL\|_F^2, \quad \text{such that } L = Z, \tag{11}$$

where λ_1, λ_2 and ρ denote the regularization parameters, Z is a slack variable, T represents a differential operator [5] such that such that $LT = [l_2 - l_1, l_3 - l_2, ..., l_N - l_{N-1}]$ and S represents the spatial relation among measurements of different sensors [5] such that SL gives the spatial difference of L [2]. Minimizing $\|LT\|_F^2$ enforces temporal consistency of the sensor readings whereas minimizing $\|SL\|_F^2$ enforces spatial correlation between sensor readings. The deep unfolded steps of ADMM-Net are then written as

$$\mathrm{vec}(L^{k+1}) = (\widetilde{U} + \lambda_1^k\widetilde{T} + \lambda_2^k\widetilde{S} + \zeta^k\widetilde{I})^{-1}\mathrm{vec}\left[\zeta^k(Z^k - P^k) + D_\Omega\right], \tag{12}$$

$$Z^{k+1} = \Psi_{\tau^k}\left\{L^{k+1} + P^k\right\}, \tag{13}$$

$$P^{k+1} = P^k + \eta^k(L^{k+1} - Z^{k+1}), \tag{14}$$

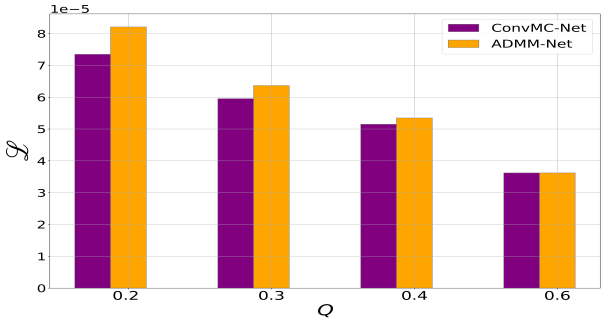

Fig. 1: NMSE with sampling rate

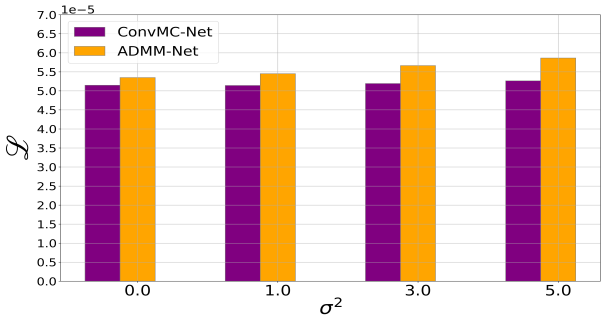

Fig. 2: NMSE with noise

where ζ is a penalty parameter introduced when making the augmented Lagrangian from (II-C), $\widetilde{I}$ is the MN dimensional identity matrix, $P = M/\zeta$ where M is the Lagrange multiplier and $\widetilde{U} = \text{diag}(\text{vec}(U))$ where U follows the definition described in (1). Furthermore, $\widetilde{S} = I_N \bigotimes \left[(S^k)^T S^k\right]$ and $\tilde{T} = (TT^T) \bigotimes I_M$, where $\bigotimes$ represents the Kronecker product. The parameters $\{S^k, \lambda_1^k, \lambda_2^k, \zeta^k, \tau^k, \eta^k\}_{k=1}^K$ are learned by the network at every layer in ADMM-Net. The loss function is the same as the one used in ConvMC-Net.

III. Experimental Evaluations on a Real-World Sensing Dataset

1) *Experiment Settings:* We compare the performance of ConvMC-Net with the deep unfolded network of ADMM-Net, both composed of 5 layers, on a real-world sensing dataset. It consists of temperature data collected every 30 seconds by 54 sensors distributed in Intel Berkeley Research Lab [15] during February 28, 2004 and April 5, 2004. 468 such groundtruth low-rank matrices L, each of size 49×60 and further corrupted by zero-mean Gaussian noise, are created from this time series dataset. Each row in L represents the successive readings of a sensor. Using a uniform probability density function, we randomly take out $Q \cdot M \cdot N$ entries from L to create the incomplete matrix D_Ω. Q is the sampling rate defined as $Q = \text{card}(\Omega)/(M.N)$, where card(.) denotes the cardinality of a set. The first 400 matrices, consisting of the first $(400 \times 49 \times 60)$ temperature recordings, are used for training both ConvMC-Net and ADMM-Net and the remaining 68 matrices are used for testing. Furthermore, both the networks are trained for 40 epochs using ADAM optimizer with a learning rate of 0.01.

2) *Results:* Fig. 1 shows the average NMSE loss $\mathcal{L}$ obtained by both ConvMC-Net and ADMM-Net as Q increases from 0.2 to 0.6. As expected, $\mathcal{L}$ decreases for both ConcMC-Net and ADMM-Net as Q increases. However for lower values of Q, which makes MC much more challenging, ConvMC-Net gives lower $\mathcal{L}$ than that given by ADMM-Net. Fig. 2 shows the loss $\mathcal{L}$ obtained by the two networks plotted against the noise power of the zero mean Gaussian noise added when preparing the dataset. We observe that as the noise power increases, the loss given by ADMM-Net also increases whereas the loss given by ConvMC-Net increases ever so marginally. This improved robustness can be attributed to the learnable kernels introduced in the unfolding process of ConvMC-Net, unlike ADMM-Net which only learns its tunable parameters from data.

TABLE I: Training and Testing time per sample for ADMM-Net and ConvMC-Net.

Method	Training time (s)	Testing time (s)
ConvMC-Net	**0.02283**	**0.01425**
ADMM-Net	0.4245	0.2629

Perhaps the biggest advantage of ConvMC-Net is that it takes significantly less time in both training and at inference than that taken by ADMM-Net as shown in Table I. This is because ADMM-Net in (12) involves computing the inverse of $MN \times MN$ sized matrix at every layer resulting in much greater computation time. This also limits ADMM-Net's applicability on high-dimensional matrices as large amount of RAM would be needed to calculate and store these inverses. The aforementioned advantages of ConvMC-Net particularly that of less training and inference time together with less computational burden makes it an attractive candidate for real-time online applications on hardware constrained resources.

IV. Conclusion

References

[1] Huynh Van Luong, B. Joukovsky, Yonina C. Eldar, and N. Deligiannis. A deep-unfolded reference-based rpca network for video foreground-background separation. *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1432–1436, 2021.

[2] Xinglin Piao, Yongli Hu, Yanfeng Sun, Baocai Yin, and Junbin Gao. Correlated spatio-temporal data collection in wireless sensor networks based on low rank matrix approximation and optimized node sampling. *Sensors*, 14(12):23137–23158, 2014.

[3] E. J. Cand'es and B. Recht. Exact matrix completion via convex optimization. *Found. Comut. Math.*, 9(6):717, 2009.

[4] Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[5] Xianghui Mao, Kai Qiu, Tiejian Li, and Yuantao Gu. Spatio-temporal signal recovery based on low rank and differential smoothness. *IEEE Transactions on Signal Processing*, 66(23):6281–6296, 2018.

[6] Xiaoke Zhu, Xiao-Yuan Jing, Di Wu, Zhenyu He, Jicheng Cao, Dong Yue, and Lina Wang. Similarity-maintaining privacy preservation and location-aware low-rank matrix factorization for qos prediction based web service recommendation. *IEEE Transactions on Services Computing*, 14(3):889–902, 2021.

[7] Nir Shlezinger, Yonina C Eldar, and Stephen P Boyd. Model-based deep learning: On the intersection of deep learning and optimization. *arXiv preprint arXiv:2205.02640*, 2022.

[8] Liu Yang, Yonina C. Eldar, Haifeng Wang, Kai Kang, and Hua Qian. An admm-net for data recovery in wireless sensor networks. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 1712–1716, 2021.

[9] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[10] Zhao Tan, Yonina C. Eldar, Amir Beck, and Arye Nehorai. Smoothing and decomposition for analysis sparse recovery. *IEEE Transactions on Signal Processing*, 62(7):1762–1774, 2014.

[11] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[12] Oren Solomon, Regev Cohen, Yi Zhang, Yi Yang, Qiong He, Jianwen Luo, Ruud J. G. van Sloun, and Yonina C. Eldar. Deep unfolded robust pca with application to clutter suppression in ultrasound. *IEEE Transactions on Medical Imaging*, 39(4):1051–1063, 2020.

[13] HanQin Cai, Jialin Liu, and Wotao Yin. Learned robust pca: A scalable deep unfolding approach for high-dimensional outlier detection. *Advances in Neural Information Processing Systems*, 34:16977–16989, 2021.

[14] Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021.

[15] Intel berkeley research lab. http://db.lcs.mit.edu/labdata/labdata.html, 2004.