DEEP GEOMETRIC MATRIX COMPLETION: A NEW WAY FOR RECOMMENDER SYSTEMS

Federico Monti, Michael M. Bronstein

Università della Svizzera italiana
Lugano, Switzerland

X. Bresson

Nanyang Technological University
Singapore

ABSTRACT

In the last years, Graph Convolutional Neural Networks gained popularity in the Machine Learning community for their capability of extracting local compositional features on signals defined on non-Euclidean domains. Shape correspondence, document classification, molecular properties predictions are just few of the many different problems where these techniques have been successfully applied. In this paper we will present Deep Geometric Matrix Completion, a recent application of Graph Convolutional Neural Networks to the matrix completion problem. We will illustrate MGCNN (a multi-graph CNN able to deal with signals defined over multiple domains) and we will show how coupling such technique with a RNN, a learnable diffusion process can be realized for reconstructing the desired information. Extensive experimental evaluation shows how Geometric Deep Learning techniques allow to outperform previous state of the art solutions on the matrix completion problem.

Index Terms— Geometric Deep Learning, Graph Convolutional Neural Networks, Matrix Completion, Recommender Systems.

1. INTRODUCTION

Convolutional Neural Networks [9, 10, 11, 12, 13] probably represent the most successful example of Deep Learning techniques nowadays. Thanks to their particular structure, CNNs allow to extract, with just a handful of parameters, local, compositional and invariant features from the signals given in input. This provides a natural way for realizing powerful models with extremely contained complexities wrt fully connected solutions (i.e. less subjected to overfitting and with better generalization capabilities). Despite the success Convolutional Neural Networks achieved in the last decade, these particular techniques have mainly been applied, until recent years, only to signals defined on Euclidean domains (i.e. grids). However, in a multitude of different fields (e.g. Biology, Computer Graphics and Social Science), one may have to deal with signals defined on non-Euclidean domains (i.e. graphs and manifolds). Unfortunately, convolution (namely the fundamental operation CNNs relies on) appears undefined on non-Euclidean domains. In order to extend CNNs to non-Euclidean structured data, in the last years several different researchers proposed possible generalized versions of convolution able to deal with signals defined over non-Euclidean domains [2,3,4,5,14,15,16,17,18]. This trend falls under the name of Geometric Deep Learning [19] and will represent one of the main topics of this particular paper.

1.1. GDL and Recommender Systems

Recovering missing information from very few measurements represents a pivotal problem for modern Recommender Systems. Products on Amazon, movies on Netflix or soundtracks on Spotify can be recommended just by reconstructing the missing entries of a sparse score matrix. In the years, several different researchers casted the matrix completion problem as a minimization task [20, 21, 22, 23, 24]. Matrix rank [20] and smoothness of rows/columns [21] over graphs where users and items are defined, represent in this sense popular constraints for achieving successful reconstructions. Despite the nice results mimization approaches achieved along the years, such techniques do not fully exploit the local stationarity structures that users/items present in corresponding graphs and the number of parameters to learn is at least linear w.r.t. the number of users and items. In 2017, Monti et al. introduced RMGCNN, the first Graph Convolutional Neural Network able to operate on the users and items graphs for recovering missing information. Such approach appears as the first attempt of applying GDL techniques to the matrix completion problem and currently represents the most prominent example of Deep Geometric Matrix Completion solutions [1,18,31]. In this paper we will revise RMGCNN. In Section 2 we will review classic matrix completion approaches. In Section 3 we will give the basics of Geometric Deep Learning and signal processing on graphs. In Section 4 we will show how GDL techniques can be applied to the matrix completion problem. In Section 5 we will finally draw conclusions.

2. MATRIX COMPLETION

The recommendation problem can be casted as a matrix completion task. Given a sparse matrix $\mathbf{X}$ where rows are items,

columns are users and entries relevance scores, the goal is to fill in the rest and thus predict the importance that each single item presents for every possible user. Unfortunately, such problem appears ill-posed without any further constraint. A well-posed version of the matrix completion problem consists in assuming matrix $\mathbf{X}$ as low-rank. Given a set Ω of known entries, whose values are represented as $y_{ij} \in \mathbf{Y}$, a reconstruction of $\mathbf{X}$ can be obtained as:

$$\arg\min_{\mathbf{X}} \ \text{rank}(\mathbf{X}) \quad \text{s.t.} \quad x_{ij} = y_{ij}, \ \forall ij \in \Omega. \tag{1}$$

Unfortunately, solving such problem turns out to be NP-hard, making thus impossible to reconstruct the missing information in most practical cases. In an attempt to reduce the computational complexity, in 2009 Candès & Recht [20] proposed to deal with the tightest possible relaxation of the rank operator. Namely, the nuclear norm. Valuable reconstructions of $\mathbf{X}$ can thus be obtained as:

$$\arg\min_{\mathbf{X}} \ \|\mathbf{X}\|_\star + \frac{\mu}{2}\|\mathbf{\Omega} \circ (\mathbf{X} - \mathbf{Y})\|_{\text{F}}^2; \tag{2}$$

the equality constraint is replaced with a penalty to make the problem robust to noise (here $\mathbf{\Omega}$ is the indicator matrix of the known entries Ω and $\circ$ denotes the Hadamard product). Cands and Recht [6] proved that any low-rank matrix can be perfectly recovered by (2), if a sufficient amount of (uniformly sampled) entries are provided.

2.1. Geometric Matrix Completion

Despite the mathematically sound approach proposed by Candès, the solution depicted in [20] ignores any kind of relationship that may exist among different users or items. A situation which appears extremely common in modern recommender systems [21, 22, 23, 24, 25]. In an attempt to include such information in the minimization problem, in 2014 Kalofolias et al. [21] proposed to reconstruct the missing information of $\mathbf{X}$ by solving:

$$\arg\min_{\mathbf{X}} \ \|\mathbf{X}\|_{\mathcal{G}_r}^2 + \|\mathbf{X}\|_{\mathcal{G}_c}^2 + \frac{\mu}{2}\|\mathbf{\Omega} \circ (\mathbf{X} - \mathbf{Y})\|_{\text{F}}^2. \tag{3}$$

Here $\mathcal{G}_r$/$\mathcal{G}_c$ denotes a row/column similarity graph connecting items/users with similar profiles. $\mathcal{G}_c = (\mathcal{V}_c, \mathcal{E}_c, \mathbf{W}_c)$ with vertex set $\mathcal{V}_c = \{1, \dots, n\}$, edge set $\mathcal{E}_c$ and adjacency matrix $\mathbf{W}_c = (w_{ij}^c)$; $w_{ij}^c = w_{ji}^c$, $w_{ij}^c = 0$ if $(i,j) \notin \mathcal{E}_c$ and $w_{ij}^c > 0$ if $(i,j) \in \mathcal{E}_c$. $\|\mathbf{X}\|_{\mathcal{G}_c}^2$ corresponds to the standard Dirichlet energy, $\|\mathbf{X}\|_{\mathcal{G}_c}^2 = \text{trace}(\mathbf{X}^\top \mathbf{\Delta}_c \mathbf{X})$, where $\mathbf{\Delta}_c$ is the graph laplacian; $\mathbf{\Delta}_c = \mathbf{I} - \mathbf{D}_c^{-1/2}\mathbf{W}_c\mathbf{D}_c^{-1/2}$, with $\mathbf{D}_c = \text{diag}(\sum_{j \neq i} w_{ij}^c)$. The same states similarly for item graph $\mathcal{G}_r$.

The main intuition is to impose smoothness over the community structure that implicitly people/items with similar tastes/features form inside similarity graphs. A result which is obtained by means of penalties $\|\mathbf{X}\|_{\mathcal{G}_r}$, $\|\mathbf{X}\|_{\mathcal{G}_c}$.

2.2. Factorized Models

In general recommender systems, one may have to deal with an enormous amount of items or customers (e.g. Netflix, Amazon...). In order to reduce the computational complexity and thus achieve efficient reconstructions, factorized representations appeared on the scene in recent years [22, 23, 25, 26, 27, 28]. The main idea is to decompose matrix $\mathbf{X}$ as the product of two low-dimensional matrices $\mathbf{W}$ and $\mathbf{H}$, respectively containing the features describing items and customers. In formula, $\mathbf{X} = \mathbf{W}\mathbf{H}^\top$. Such representation appears particular attractive, since it allows to reduce the number of features to learn from $O(n \cdot m)$ to $O(n + m)$. The reduced model complexity allows in addition to reduce overfitting, improving the quality of the reconstructed entries (see [22] in Table 2). Problems (2) and (3) boil down to:

$$\arg\min_{\mathbf{W},\mathbf{H}} \ \frac{1}{2}\|\mathbf{W}\|_{\text{F}}^2 + \frac{1}{2}\|\mathbf{H}\|_{\text{F}}^2 + \frac{\mu}{2}\|\mathbf{\Omega} \circ (\mathbf{W}\mathbf{H}^\top - \mathbf{Y})\|_{\text{F}}^2, \tag{4}$$

$$\arg\min_{\mathbf{W},\mathbf{H}} \ \frac{1}{2}\|\mathbf{W}\|_{\mathcal{G}_r}^2 + \frac{1}{2}\|\mathbf{H}\|_{\mathcal{G}_c}^2 + \frac{\mu}{2}\|\mathbf{\Omega} \circ (\mathbf{W}\mathbf{H}^\top - \mathbf{Y})\|_{\text{F}}^2. \tag{5}$$

3. GEOMETRIC DEEP LEARNING

Graph Convolutional Neural Networks are rapidly becoming popular solutions for solving prediction tasks on non-Euclidean structured data [2, 3, 5, 14, 15, 16, 17, 18]. Due to space limitations, in this section we will focus on spectral Graph Convolutional Neural Networks, which represent valuable solutions for signals defined over a single domain.

Given a generic graph $\mathcal{G}$ defined by adjacency matrix $\mathbf{W}$ and given $\mathbf{\Delta} = \mathbf{D} - \mathbf{W} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Phi}^\top$ the corresponding graph laplacian, we can define convolution in the spectral domain as $x \star y = \mathbf{\Phi}(\mathbf{\Phi}^\top x) \cdot (\mathbf{\Phi}^\top y) = \mathbf{\Phi}(\hat{x} \cdot \hat{y})$. The eigenfunctions $\mathbf{\Phi}$ of the graph laplacian play in this case the role of Fourier atoms (thus $\mathbf{\Phi}^\top x$ correspond to a Graph Fourier Transform) and the corresponding eigenvalues $\mathbf{\Lambda}$ represent frequencies (the smoother $\boldsymbol{\phi}_i$ the smaller λ_i, with $\lambda_i \geq 0 \ \forall \ i$).

Bruna et al. [14] exploited this formulation for defining graph convolutional layers in the form:

$$\tilde{\mathbf{x}}_l = \xi\left(\sum_{l'=1}^{q'} \mathbf{\Phi}\hat{\mathbf{Y}}_{ll'}\mathbf{\Phi}^\top \mathbf{x}_{l'}\right), \quad l = 1, \dots, q, \tag{6}$$

where q', q denote the number of input and output channels, $\hat{\mathbf{Y}}_{ll'} = \text{diag}(\hat{y}_{ll',1}, \dots, \hat{y}_{ll',n})$ is a diagonal matrix of spectral multipliers and ξ is a nonlinearity (e.g. ReLU). Unfortunately, this particular formulation requires a number of parameters which is linear wrt the number of vertices n available in the given graph, doesn't admit efficient computations (since it requires projections of input signals $\mathbf{x}_{l'}$ over dense matrix $\mathbf{\Phi}$) and doesn't guarantee localized filters over the given domain.

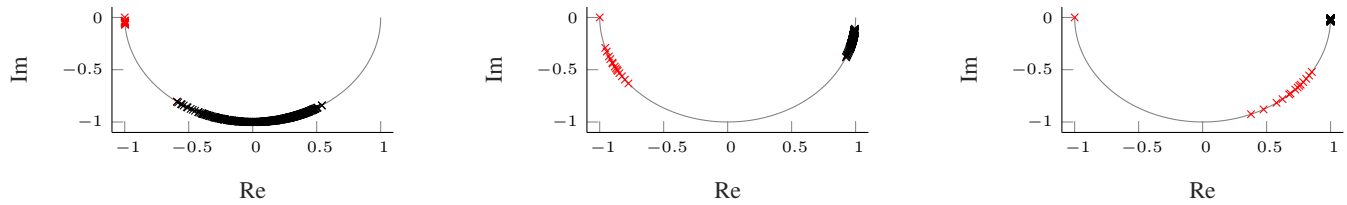

Fig. 1. Eigenvalues of the unnormalized Laplacian $h\boldsymbol{\Delta}$ of a 15-communities graph mapped by means of Cayley transform with spectral zoom values (left-to-right) $h = 0.1$, 1, and 10. The first 15 frequencies carrying most of the information about the communities are marked in red. Note how the distance between corresponding eigenvalues changes for different values of h.

In order to contain the amount of parameters and guarantee, at least in some sense, localized filters, in [14] Bruna et al. (and in a follow-up work Henaff et al. [15]) argued that local filters can be achieved by means of smooth spectral transfer functions. The spectral multipliers are defined in this case as $\hat{y}_k = \tau(\lambda_k) = \sum_{j=1}^{p} \theta_j \beta_j(\lambda_k)$, where $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_p)$ correspond to the parameters to learn and $\boldsymbol{\beta} = (\beta_1(\lambda), \beta_2(\lambda), \dots, \beta_p(\lambda))$ is just a set of interpolation kernels. The computational complexity of such method unfortunately remains a $O(n^2)$.

In 2016, Defferrard et al. proposed to exploit Chebyshev polynomials for reducing the computational complexity of spectral GCNs [16]. Defining filters as polynomials applied over the eigenvalues of the graph laplacian, it is possible indeed to avoid any eigendecomposition and realize convolution by means of efficient sparse routines (the computational complexity drops in this way from $O(n^2)$ to $O(p \cdot |\mathcal{E}|)$). Convolution with Chebyshev polynomials can be obtained as:

$$\tilde{\mathbf{x}} = \boldsymbol{\Phi} \sum_{j=0}^{p} \theta_j T_j(\tilde{\lambda}) \boldsymbol{\Phi}^\top \mathbf{x} = \sum_{j=0}^{p} \theta_j T_j(\tilde{\boldsymbol{\Delta}}) \mathbf{x}, \qquad (7)$$

where $\tilde{\lambda}$ is a frequency rescaled in $[-1, 1]$, $\boldsymbol{\theta}$ is the $(p+1)$-dimensional vector of polynomial coefficients parametrizing the filter and $T_j(\lambda) = 2\lambda T_{j-1}(\lambda) - T_{j-2}(\lambda)$ denotes the Chebyshev polynomial of degree j ($T_1(\lambda) = \lambda$ and $T_0(\lambda) = 1$). Here, $\tilde{\boldsymbol{\Delta}} = 2\lambda_n^{-1}\boldsymbol{\Delta} - \mathbf{I}$ is the rescaled Laplacian with eigenvalues $\tilde{\boldsymbol{\Lambda}} = 2\lambda_n^{-1}\boldsymbol{\Lambda} - \mathbf{I}$ in the interval $[-1, 1]$.

Despite the efficiency presented by ChebNet [16], the approach proposed by Defferrard et al. struggles at dealing with graphs presenting contracted spectrums e.g. community graphs [18] . An alternative solution is thus represented by CayleyNet [18]. The main idea behind CayleyNet is to achieve some sort of spectral zoom property by means of Cayley transform. Applying the Cayley transform to rescaled eigenvalues of the graph laplacian $\boldsymbol{\Delta}$ (i.e. $C(h\lambda) = (h\lambda - i)(h\lambda + i)^{-1}$) results indeed in a non-linear transformation of the eigenvalues (Figure 1). This provides a natural way for zooming on some specific frequencies, which may be the most relevant for the considered classification task. Convolution with Cayley filters can be obtained as:

$$\tilde{\mathbf{x}} = c_0 + 2\mathrm{Re}\Big\{ \sum_{j=1}^{p} c_j C(h\boldsymbol{\Delta})^j \Big\} \mathbf{x}, \qquad (8)$$

where $\mathbf{c} = (c_1, c_2, \dots, c_p)$ is a set of complex coefficients and $C(h\boldsymbol{\Delta}) = (h\boldsymbol{\Delta} - i\mathbf{I})(h\boldsymbol{\Delta} + i\mathbf{I})^{-1}$. Matrix inversion can be avoided recursively rewriting the projections of signal $\mathbf{x}$ over the powers of the transformed laplacian and approximately solving the obtained linear systems of equations by means of Jacobi method [18] (the final computational complexity is a $O(K \cdot p \cdot |\mathcal{E}|)$, where K is the number of Jacobi iterations).

4. DEEP GEOMETRIC MATRIX COMPLETION

4.1. Multigraph Convolutions

Following what proposed by Kalofolias et al. [21] (and subsequently in [22, 25]), it's easy to observe how better matrix completions can be achieved by considering the sparse matrix $\mathbf{X}$ as defined over two different graphs: a user graph and an item graph. From a signal processing point of view, matrix $\mathbf{X}$ can be considered as a bi-dimensional signal defined over two distinct domains. In order to apply GCNs to this particular situation, an extension of the aforementioned Graph Fourier Transform needs to be introduced for signals defined over multiple graphs. Recalling that a 2D Fourier transform can be achieved by applying two independent mono-dimensional Fourier transforms on the rows and columns of the provided matrix, a multi-grah Fourier trasform can be obtained as: $\hat{\mathbf{X}} = \boldsymbol{\Phi}_r^\top \mathbf{X} \boldsymbol{\Phi}_c$; where $\boldsymbol{\Phi}_c, \boldsymbol{\Phi}_r$ denote the $n \times n$ and $m \times m$ eigenvector matrices of the column- and row-graph Laplacians $\boldsymbol{\Delta}_c, \boldsymbol{\Delta}_r$. Defining a multi-graph convolution is then straightforward, given the spectrum $\hat{\mathbf{X}}$ of the provided matrix $\mathbf{X}$, a multi-graph convolution can be obtained just by means of an element-wise multiplication with some set of spectral multipliers: $\mathbf{X} \star \mathbf{Y} = \boldsymbol{\Phi}_r(\hat{\mathbf{X}} \circ \hat{\mathbf{Y}})\boldsymbol{\Phi}_c^\top$. Following what presented in [16, 18], in order to realize efficient multi-graph convolutions, Monti et al. [1] proposed to parametrize the spectral multipliers as the product of two independent Chebyshev polynomials: $\tau_{\boldsymbol{\Theta}}(\tilde{\lambda}_c, \tilde{\lambda}_r) = \sum_{j,j'=0}^{p} \theta_{jj'} T_j(\tilde{\lambda}_c) T_{j'}(\tilde{\lambda}_r)$. Heavy eigendecompositions and projections over the eigenfunctions of the graph laplacians are thus avoided.

4.2. Separable Convolutions

Despite the nice idea illustrated in the previous subsection, the approach outlined in 4.1 requires a $O(n \cdot m)$ number

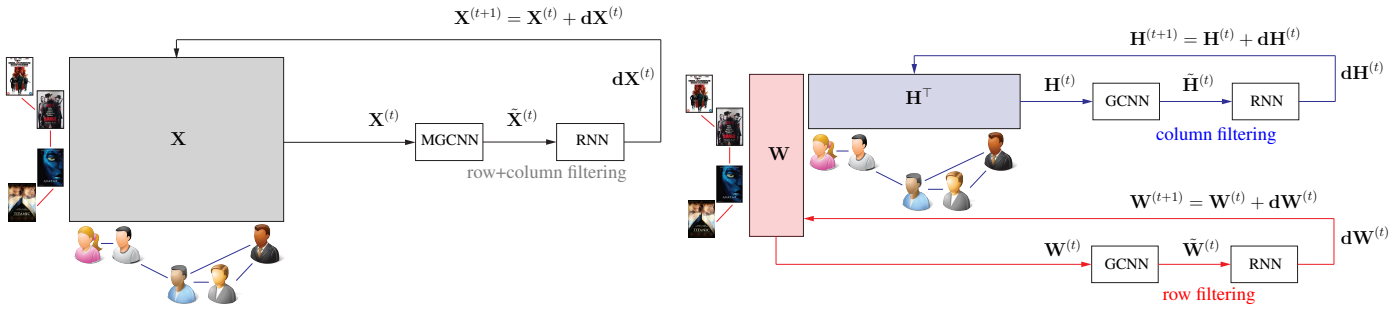

Fig. 2. Recurrent GCNN (RGCNN) architecture using the multi-graph (left) and separable (right) multi-graph convolution. The number of parameters to learn is $\mathcal{O}(1)$ and the learning complexity is respectively $\mathcal{O}(n \cdot m)$ and $\mathcal{O}(n + m)$.

of operations. This makes difficult to apply such solution to really large and huge graphs (e.g. in the famous Netflix challenge [27] people had to deal with a matrix representing 480k movies and 18k users i.e. 8.5B entries in total). A solution to the problem is represented by factorized approaches [1, 18]. Instead of recurring to multi-graph convolutions realized over the entire matrix $\mathbf{X}$, two independent single-graph GCNs can be applied on matrices $\mathbf{W}$ and $\mathbf{H}$: $\tilde{\mathbf{w}}_l = \xi\left(\sum_{l'=1}^{q'} \sum_{j=0}^{p} \theta^r_{j,ll'} T_j(\tilde{\mathbf{\Delta}}_r)\mathbf{w}_{l'}\right)$, $\tilde{\mathbf{h}}_l = \xi\left(\sum_{l'=1}^{q'} \sum_{j'=0}^{p} \theta^c_{j',ll'} T_{j'}(\tilde{\mathbf{\Delta}}_c)\mathbf{h}_{l'}\right)$. This allows to reduce the amount of operations to $O(n + m)$, giving the possibility to scale the presented architecture to extremely large matrices.

4.3. Matrix completion as a diffusion process

Given the aforementioned multi-graph convolutional layers, the last step that remains concerns the choice of the architecture to use for reconstructing the missing information. In [1] the authors proposed to model matrix completion as a diffusion process by means of a RNN casted on top of the aforementioned GCNs (Figure 2). Every (user, item) pair in the multi-graph approach and every user/item in the separable one present in this case an independent state, which is updated (at every step) by means of the features produced by the selected GCN. Such state is used for predicting an incremental update $dx^t_{i,j}$ (dw^t_i, dh^t_j for the separable solution) that is summed to the current value $x^t_{i,j}$ (w^t_i, h^t_j) for reconstructing the missing entries. Experimental evaluation with Chebyshev polynomials (order $p = 4$ and $T = 10$ diffusion steps) on synthetic datasets shows how this particular solution (RMGCNN) outperforms deeper MGCNNs on the matrix completion task (Table 1).

On real datasets, Geometric Deep Learning techniques consistently outperform classic matrix completion approaches, achieving state of the art performance on the Movielens $100K$, Flixster, Douban and Yahoo Music dataset[1] (Table 2). Architectures based on Cayley polynomials appear in particular as better solutions, thanks to the spectral zoom properties they present (which allow to handle possible communities without incurring in excessive polynomial orders).

[1]Submatrices of 3000 $\times$ 3000 entries have been extracted for Flixster, Douban and Yahoo Music, in order to reduce the computational complexity.

Table 1. RMSE obtained on a synthetic dataset. The first hidden layer receiving in input matrix $\mathbf{X}$ and producing in ouput 32 features for each (user, item) pair (i.e. 1MGC32) is omitted for reasons of space.

Method	Params	Architecture	RMSE
MGCNN_{3L}	$9K$	32MGC10, 10MGC1	0.0116
MGCNN_{4L}	$53K$	32MGC32 $\times$ 2, 32MGC1	0.0073
MGCNN_{5L}	$78K$	32MGC32 $\times$ 3, 32MGC1	0.0074
MGCNN_{6L}	$104K$	32MGC32 $\times$ 4, 32MGC1	0.0064
RMGCNN	**9K**	LSTM	**0.0053**

Table 2. Performance (RMSE) of several matrix completion methods on the MovieLens, Flixster, Douban and Yahoo Music datasets.

Method	**MovieLens**	**Flixster**	**Douban**	**Yahoo**
IMC [29, 30]	1.653	–	–	–
GMC [21]	0.996	–	–	–
MC [20]	0.973	–	–	–
GRALS [22]	0.945	1.245	0.833	38.042
$\text{sRMGCNN}_{\text{Cheby}}$ [1]	**0.929**	**0.926**	**0.801**	**22.415**
$\text{sRMGCNN}_{\text{Cayley}}$ [18]	**0.922**	–	–	–

5. CONCLUSIONS

GDL techniques are rapidly becoming popular solutions for solving all sort of prediction tasks on non-Euclidean structured data. In this paper we reviewed Deep Geometric Matrix Completion, a recent application of these techniques to the matrix completion problem. We showed how spectral Graph Convolutional Neural Networks can be used to achieve successful reconstructions, while keeping at the same time contained computational burdens. Despite the nice performance achieved, the works here illustrated only represent the first attempts of applying GDL techniques to the recommendation problem. We believe new approaches will exponentially appear in the literature (e.g. [31]), showing how GCNs can effectively be used for further improving the results here presented.

6. REFERENCES

[1] F. Monti, M. M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," 2017.

[2] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst, "Geodesic convolutional neural networks on Riemannian manifolds," in *Proc. 3DRR*, 2015.

[3] D. Boscaini, J. Masci, E. Rodolà, and M. M. Bronstein, "Learning shape correspondence with anisotropic convolutional neural networks," in *Proc. NIPS*, 2016.

[4] O. Litany, T. Remez, E. Rodolà, A. M. Bronstein, and M. M. Bronstein, "Deep functional maps: Structured prediction for dense shape correspondence," *arXiv preprint arXiv:1704.08686*, 2017.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv:1609.02907*, 2016.

[6] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," *arXiv:1511.02136v2*, 2016.

[7] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. NIPS*, 2015.

[8] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, "Molecular graph convolutions: moving beyond fingerprints," *Journal of computer-aided molecular design*, vol. 30, no. 8, pp. 595–608, 2016.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012.

[11] M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR*, vol. abs/1312.4400, 2013.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv:1512.03385*, 2015.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[14] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *Proc. ICLR*, 2013.

[15] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv:1506.05163*, 2015.

[16] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. NIPS*, 2016.

[17] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model CNNs," in *Proc. CVPR*, 2017.

[18] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters," *arXiv preprint arXiv:1705.07664*, 2017.

[19] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *arXiv:1611.08097*, 2016.

[20] E. Candès and B. Recht, "Exact Matrix Completion via Convex Optimization," *Foundations of Computational Mathematics*, vol. 9(6), pp. 717–772, 2009.

[21] V. Kalofolias, X. Bresson, M. M. Bronstein, and P. Vandergheynst, "Matrix completion on graphs," *arXiv:1408.1717*, 2014.

[22] N. Rao, H.-F. Yu, P. K. Ravikumar, and I. S. Dhillon, "Collaborative filtering with graph information: Consistency and scalable methods," in *Proc. NIPS*, 2015.

[23] H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. Web Search and Data Mining*, 2011.

[24] D. Kuang, Z. Shi, S. Osher, and A. Bertozzi, "A harmonic extension approach for collaborative ranking," *arXiv:1602.05127*, 2016.

[25] K. Benzi, V. Kalofolias, X. Bresson, and P. Vandergheynst, "Song recommendation with non-negative matrix factorization and graph total variation," in *Proc. ICASSP*, 2016.

[26] F. Yanez and F. Bach, "Primal-dual algorithms for non-negative matrix factorization with the kullback-leibler divergence," *arXiv:1412.1788*, 2012.

[27] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[28] N. Srebro, J. Rennie, and T. Jaakkola, "Maximum-Margin Matrix Factorization," in *Proc. NIPS*, 2004.

[29] P. Jain and I. S. Dhillon, "Provable inductive matrix completion," *arXiv:1306.0626*, 2013.

[30] M. Xu, R. Jin, and Z.-H. Zhou, "Speedup matrix completion with side information: Application to multi-label learning," in *Proc. NIPS*, 2013.

[31] M. W. Rianne van den Berg, Thomas N. Kipf, "Graph convolutional matrix completion," *arXiv preprint arXiv:1706.02263*, 2017.