Learning Connectivity with Graph Convolutional Networks

Hichem SAHBI

CNRS, Sorbonne University, Paris, France

ICPR 2020

Motivation and Contribution

Motivation

- Graph convolutional networks (GCNs) aim at generalizing deep learning to arbitrary irregular domains.
- The general principle of spatial GCNs consists in aggregating node representations before applying convolution to node aggregates.
- The success of spatial GCNs is reliant on the topology (or structure) of input graphs.
- However, graph structures (either available or handcrafted) are powerless to optimally capture all the relationships between nodes as their setting is oblivious to the targeted applications.
- E.g., node-to-node relationships, in human skeletons, capture the intrinsic anthropometric characteristics of individuals (useful for their identification) while other connections, yet to infer, are necessary for recognizing their dynamics and actions.

Contribution

- We introduce a novel framework that learns convolutional filters on graphs together with their topological properties.
- The latter are modeled through matrix operators that capture multiple aggregates on graphs, learned using a constrained cross-entropy loss.
- We consider different *constraints* (including stochasticity, orthogonality and symmetry) acting as regularizers which reduce the space of possible solutions and overfitting.
- Stochasticity implements random walk Laplacians while orthogonality models multiple aggregation operators with non-overlapping supports; it also avoids redundancy and oversizing the learned GCNs with useless parameters. Symmetry reduces further the number of training parameters.

Spatial graph convolutional networks at a glance

- Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote a graph endowed with (i) a signal $\{\psi(u) \in \mathbb{R}^s\}_u$ and (ii) an adjacency matrix $\mathbf{A}$. The spatial convolution of $\mathcal{G}$ with a set of filters $\mathcal{F}$ and nodes $\mathcal{V}$ is

$$(\mathcal{G} \star \mathcal{F})_{\mathcal{V}} = f\big(\mathbf{A}\ \mathbf{U}^\top\ \mathbf{W}\big).$$

- Here $\mathbf{A}\mathbf{U}^\top$ acts as a feature extractor which collects non-differential and differential statistics including means $\{\mathbb{E}(\psi(\mathcal{N}_r(u)))\}_u$ and (up to a squared power) variances $\{\psi(u) - \mathbb{E}(\psi(\mathcal{N}_r(u)))\}_u$ of node neighbors, before applying convolutions using $\mathbf{W}$.

Learning connectivity with GCNs

Problem statement

- Considering E as the cross entropy loss, we turn the design of the connectivity matrix $\mathbf{A}$ as a part of GCN learning.
- One may use the chain rule in order to derive the gradient $\frac{\partial E}{\partial \mathbf{vec}(\mathbf{A})}$ and hence update $\mathbf{A}$ using SGD.
- We upgrade SGD by learning both the convolutional parameters of GCNs together with connectivity matrices while implementing *orthogonality, stochasticity and symmetry*.
- Orthogonality allows designing these connectivity matrices with a minimum number of parameters, stochasticity normalizes nodes by their degrees and allows learning random walk Laplacians, while symmetry reduces further the number of training parameters.

Stochasticity

- Stochasticity requires adding equality and inequality constraints in SGD, i.e., $\mathbf{A}_{ij} \in [0, 1]$ and $\sum_q \mathbf{A}_{qj} = 1$.
- We consider a reparametrization of the learned matrices, as $\mathbf{A}_{ij} = h(\hat{\mathbf{A}}_{ij})/\sum_q h(\hat{\mathbf{A}}_{qj})$, with $h : \mathbb{R} \to \mathbb{R}^+$ being strictly monotonic and this allows a free setting of the matrix $\hat{\mathbf{A}}$ during optimization while guaranteeing $\mathbf{A}_{ij} \in [0, 1]$ and $\sum_q \mathbf{A}_{qj} = 1$.
- During backpropagation, the gradient of the loss E (now w.r.t $\hat{\mathbf{A}}$) is updated using the chain rule as

$$\frac{\partial E}{\partial \hat{\mathbf{A}}_{ij}} = \sum_p \frac{\partial E}{\partial \mathbf{A}_{pj}} . \frac{\partial \mathbf{A}_{pj}}{\partial \hat{\mathbf{A}}_{ij}}.$$

- In practice $h(.) = \exp(.)$ and the new gradient (w.r.t $\hat{\mathbf{A}}$) is obtained by multiplying the original one by the Jacobian $\mathbf{J}_{\text{stc}} = \big[\frac{\partial \mathbf{A}_{pj}}{\partial \hat{\mathbf{A}}_{ij}}\big]_{p,i=1}^{n}$.

Orthogonality

- Learning multiple $\{\mathbf{A}_k\}_k$ allows us to capture different graph topologies when achieving aggregation and convolution. With multiple $\{\mathbf{A}_k\}_k$ convolution is updated as

$$(\mathcal{G} \star \mathcal{F})_{\mathcal{V}} = f\Big(\sum_{k=1}^{K} \mathbf{A}_k \mathbf{U}^\top \mathbf{W}_k\Big).$$

- Provided that $\{\psi(u')\}_{u' \in \mathcal{N}_k(u)}$ are linearly independent (l.i.), the sufficient condition that makes the aggregated representations l.i. is orthogonality, i.e., $\langle \mathbf{A}_k, \mathbf{A}_{k'} \rangle_F = \mathbf{tr}(\mathbf{A}_k^\top \mathbf{A}_{k'}) = 0$ and $\mathbf{A}_k, \mathbf{A}_{k'} \geq \mathbf{0}_n, \forall k \neq k'$, with $\langle , \rangle_F$ being the Hilbert-Schmidt (Frobenius) inner product.
- This equates (see the paper) $\mathbf{A}_k \odot \mathbf{A}_{k'} = \mathbf{0}_n, \forall k \neq k'$ with $\odot$ denoting the entrywise hadamard product and $\mathbf{0}_n$ the $n \times n$ null matrix.
- Hence, we learn the matrices as

$$\begin{array}{ll} \min_{\{\mathbf{A}_k\}_k, \mathbf{W}} & E\big(\mathbf{A}_1, \ldots, \mathbf{A}_K; \mathbf{W}\big) \\ \text{s.t.} & \mathbf{A}_k \odot \mathbf{A}_k > \mathbf{0}_n \\ & \mathbf{A}_k \odot \mathbf{A}_{k'} = \mathbf{0}_n \qquad \forall k, k' \neq k. \end{array}$$

- We investigate a workaround that optimizes these matrices while guaranteeing their orthogonality during optimization.
- We consider $\exp(\gamma \hat{\mathbf{A}}_k) \oslash (\sum_{r=1}^{K} \exp(\gamma \hat{\mathbf{A}}_r))$ as a soft/crispmax reparametrization of $\mathbf{A}_k$, with $\oslash$ being the entrywise hadamard division and $\{\hat{\mathbf{A}}_k\}_k$ free parameters in $\mathbb{R}^{n \times n}$.
- By choosing a large value of γ, it becomes possible to implement ϵ-orthogonality; a surrogate property where only one entry $\mathbf{A}_{kij} \gg 0$ while all others $\{\mathbf{A}_{k'ij}\}_{k' \neq k}$ vanish.
- The setting of γ and updated Jacobians are in the paper.

Symmetry and combination

- Symmetry is guaranteed by considering the reparametrization of each matrix as $\mathbf{A}_k = \frac{1}{2}(\hat{\mathbf{A}}_k + \hat{\mathbf{A}}_k^\top)$ with $\hat{\mathbf{A}}_k$ being a free matrix, and it is maintained by multiplying the original gradient $\frac{\partial E}{\partial \mathbf{vec}(\{\mathbf{A}_k\}_k)}$ by the Jacobian

$$\mathbf{J}_{\text{sym}} = \frac{1}{2}\big[1_{\{k=k'\}} . 1_{\{(i=i',j=j') \vee (i=j',j=i')\}}\big]_{ijk,i'j'k'}$$

which is extremely sparse and highly efficient to evaluate.

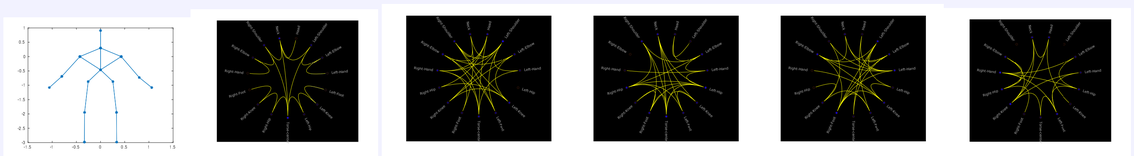
- One may combine symmetry with all the aforementioned constraints by multiplying the underlying Jacobians, so the final gradient is obtained by multiplying the original one as

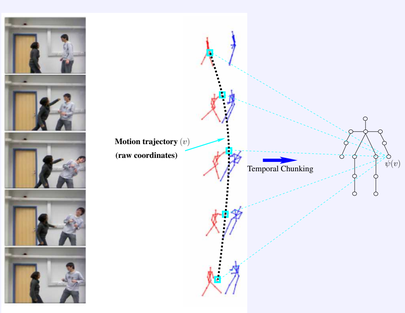
$$\frac{\partial E}{\partial \mathbf{vec}(\{\hat{\mathbf{A}}_k\}_k)} = \mathbf{J}_{(\text{sym or stc})} . \mathbf{J}_{\text{orth}} . \frac{\partial E}{\partial \mathbf{vec}(\{\mathbf{A}_k\}_k)}.$$

Experiments

- **Evaluation Set (SBU):** 282 skeleton sequences acquired using the Microsoft Kinect sensor belonging to 8 categories. Skeleton representation is based on temporal chunking.

Oper \ Const		none	sym	orth	stc	sym+orth	orth+stc	Mean
HPM.	$K=1$	89.2308	92.3077	–	89.2308	–	–	90.2564
	$K=4$	87.6923	89.2308	89.2308	87.6923	90.7692	92.3077	89.4872
	$K=8$	90.7692	95.3846	92.3077	90.7692	92.3077	92.3077	92.3077
	Mean	89.2308	92.3077	90.7692	89.2308	91.5384	92.3077	90.7692
LPM.	$K=1$	92.3077	87.6923	–	95.3846	–	–	91.7949
	$K=4$	92.3077	92.3077	93.8462	95.3846	90.7692	96.9231	93.5897
	$K=8$	95.3846	90.7692	87.6923	93.8462	93.8462	92.3077	92.3077
	Mean	93.3333	90.2564	90.7692	94.8718	92.3077	94.6154	92.7180
Our	$K=1$	95.3846	93.8462	–	95.3846	–	–	94.8718
	$K=4$	93.8462	95.3846	95.3846	96.9231	93.8462	**98.4615**	95.6410
	$K=8$	92.3077	93.8462	95.3846	90.7692	95.3846	90.7692	93.0769
	Mean	93.8462	94.3590	95.3846	94.3590	94.6154	94.6154	94.4615





Method	Accuracy
GCNConv (Kipf et al. ICLR 2017)	90.00
ArmaConv (Bianchi et al, Arxiv 2019)	96.00
SGCConv (Wu et al, Arxiv 2019)	94.00
ChebyNet (Defferrard et al., NIPS 2016)	96.00
Raw coordinates (Yun et al., CVPR 2012)	49.7
Joint features (Yun et al., CVPR 2012)	80.3
Interact Pose (Ji et al., ICMEW 2014)	86.9
CHARM (Li et ak, ICCV 2015)	83.9
HBRNN-L (Du et al, CVPR 2015)	80.35
Co-occurence LSTM (Zhu et al. AAAI 2016)	90.41
ST-LSTM (Liu et al. ECCV 2016)	93.3
Topological pose ordering (Baradel et al. Arxiv 2017)	90.5
STA-LSTM (Song et al., AAAI 2017)	91.51
GCA-LSTM (Liu et al., IEEE TIP2018)	94.9
VA-LSTM (Zhang et al., ICCV 2017)	97.2
DeepGRU (Maghoumi et al., Arxiv 2018)	95.7
Riemannian manifold trajectory (Kacem et al., IEEE PAMI 2018)	93.7
Our best model	98.43