Classifying Signals on Irregular Domains via Convolutional Cluster Pooling

Angelo Porrello, Davide Abati, Simone Calderara, Rita Cucchiara
Department of Engineering, University of Modena and Reggio Emilia, Modena, Italy
angelo.porrello@unimore.it

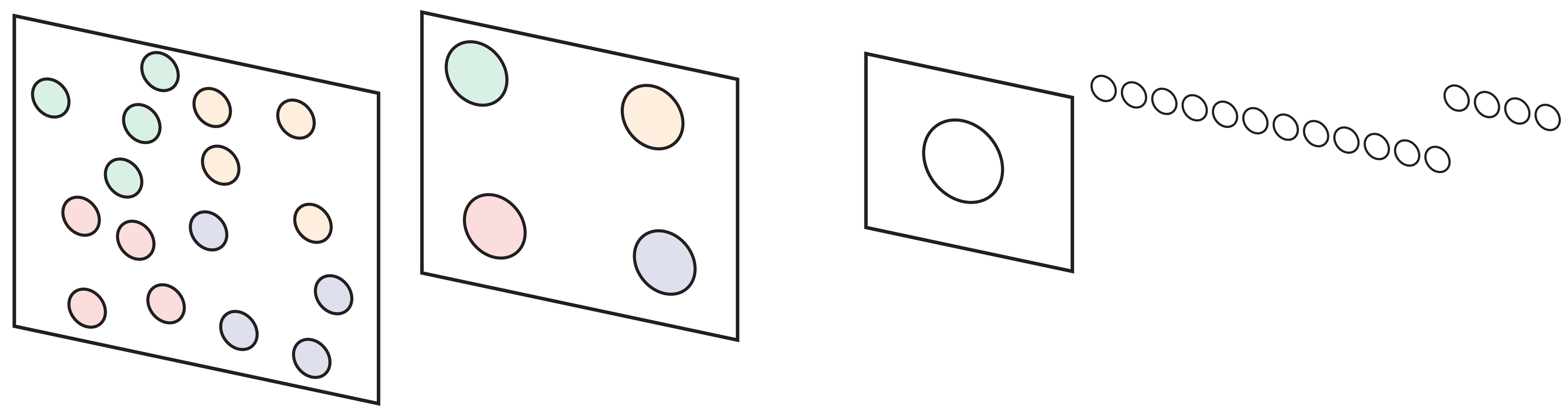
1. Motivations

We are surrounded by data lying on an underlying non-euclidean structure (e.g. 3D skeleton data, social networks and chemical compounds). Graph Convolutional Networks (GCNs) [1, 2, 3] provide a comprehensive and solid framework for *vertex classification* in such data domains, because they model directly the topological structures through edge weights. Differently, we focus on *graph signal classification*.
Our approach, built by stacking multiple **Convolutional Cluster Pooling** (**CCP**) layers, provides:

- a **hierarchical framework** for supervised learning in homogeneous graph contexts.
- a spatial formulation for graph filtering which, as for CNNs, exploits **weight sharing** and produces a pooled graph signal.

2. Our Proposal

Multiple applications of the CCP layer lead to multi-scale representations of the graph.



- *Firstly*, CCP performs a **clustering** operation on the input graph, resulting in a coarser output graph, whose affinity matrix reflects relationships among clusters regressed at training time.
- *Secondly*, the layer selects for each cluster a fixed number of candidate nodes for the **aggregation phase**, and sorts them depending on a **centrality-based rank** within the cluster.

3. Hierarchical Soft Clustering

A cascade of M soft-partitions, described by an ordered sequence of assignment matrices $K^{(1)}, K^{(2)}, \ldots, K^{(M)}$, forms a soft dendrogram for the original graph $\mathcal{A}$. The problem of obtaining a good dendrogram is formalised as follows:

$$\max_{K^{(i)} i=1,\ldots,M} \mathcal{L}_{\mathcal{K}} = \frac{1}{2} \sum_{m=1}^{M} \sum_{k=1}^{|\mathcal{K}_m|} \frac{Cohesion\,(K_k^{(m)})}{Vol(K_k^{(m)})}$$

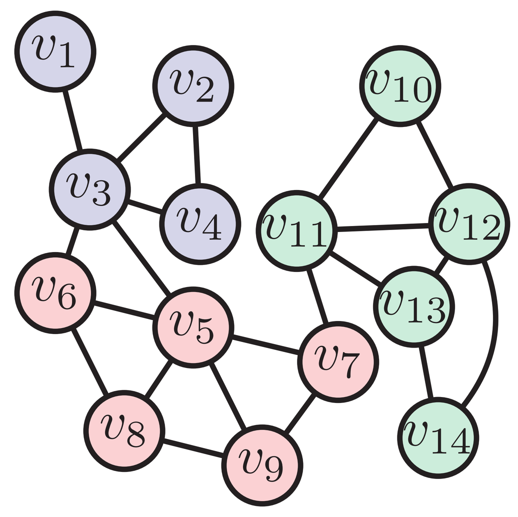
$$\text{subject to} \quad \sum_{k=1}^{|\mathcal{K}_m|} K_{i,k}^{(m)} = 1 \quad \substack{i=1,2,\ldots,|\mathcal{K}_{m-1}| \\ m=1,2,\ldots,M.}$$

This way, we require intermediate clusters with maximal cohesion and minimum size. $\mathcal{L}_{\mathcal{K}}$ is paired to the the classification loss $\mathcal{L}_0$, delivering the regularised objective $\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_{\mathcal{K}}$. This way, the supervision signal may provide information to the process of clusters formation.

4. Neighbourhood Selection

For each cluster we select as candidate set $\mathcal{N}_k^{(m+1)}$ for the filtering stage the set containing the most L representative nodes.

We consider a node more **central** if: i) it has a high membership value for the cluster under consideration; ii) a large part of its direct neighbours nodes share the same cluster in the graph.

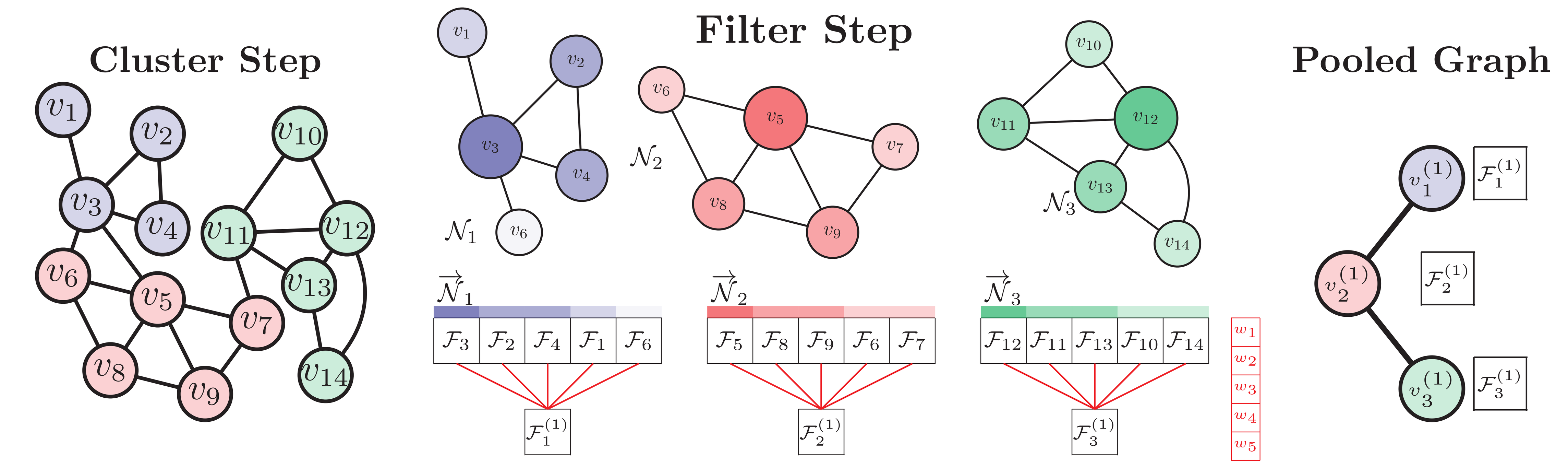


$$Rank\;(\mathcal{V}_i^{(m)} \rightarrow \mathcal{K}_k^{(m+1)}) = (1 + K_{i,k}^{(m+1)}) \sum_{j \neq i}^{|\mathcal{K}_m|} \mathcal{A}_{i,j}^{\mathcal{K}_m} K_{j,k}^{(m+1)}$$

$\overrightarrow{\mathcal{N}}_1$ $\mathcal{V}_i$	Rank	$\overrightarrow{\mathcal{N}}_2$ $\mathcal{V}_i$	Rank	$\overrightarrow{\mathcal{N}}_3$ $\mathcal{V}_i$	Rank
v_3	6	v_5	8	v_{12}	8
v_2	4	v_8	6	v_{11}	6
v_4	4	v_9	6	v_{13}	6
v_1	2	v_6	4	v_{10}	4
v_6	1	v_7	4	v_{14}	4

5. Convolutional Cluster Pooling (CCP)

The m-th CCP layer takes in input an affinity matrix $\mathcal{A}^{\mathcal{K}_m}$ and a multi-dimensional $\mathcal{F}^{(m)} \in \mathbb{R}^{|\mathcal{K}_m| \times d_{IN}}$ signal defined on the vertex set.



CCP yields:

- $\mathcal{A}^{\mathcal{K}_{m+1}}$: a new reduced affinity matrix
- $\mathcal{F}^{(m+1)} \in \mathbb{R}^{|\mathcal{K}_{m+1}| \times d_{OUT}}$: a pooled signal

$$\mathcal{F}_{k,j}^{(m+1)} = \sum_{i=1}^{d_{IN}} \sum_{l=1}^{L} W_{l,i,j}\,(\sigma_{k,l} \cdot \overrightarrow{\mathcal{N}}_k^{(m+1)}(l,i)) + b_j$$

where $W \in \mathbb{R}^{L \times d_{IN} \times d_{OUT}}$ and $b \in \mathbb{R}^{d_{OUT}}$ are parameters of our CCP layer.

6. GCN Baselines

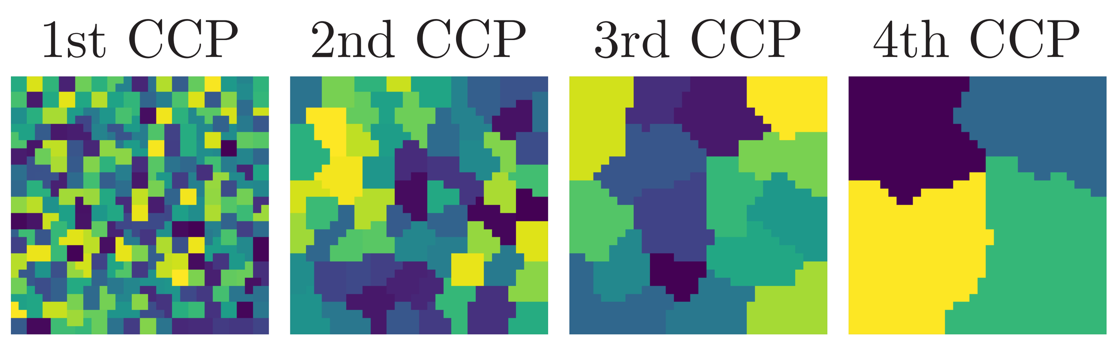
Comparison w.r.t other graph corsening and filtering approches.

Filter	Coarsen	CIFAR-10	NTU-CS
Chebyshev [1]	Graclus	78.15	74.85
GCN [2]	Graclus	67.01	62.00
GAT [3]	Graclus	72.82	59.48
CCP	**CCP**	**84.4**	**80.1**

7. Classification Results

Action recognition NTU RGB+D Method	CS Acc.	Image classification CIFAR-10 Method	Acc.	Text categorisation 20NEWS Method	Acc.
P-LSTM	62.9	Graph-CNNs	68.3	Linear SVM	65.9
TGCNN	71.4	FC	78.6	FC2500-FC500	65.8
Deep STGC$_K$	74.9	**CCP**	**84.4**	Softmax	66.3
C-CNN	79.6	Stochastic Pooling	84.9	Chebyshev - GC32	68.3
CCP	**80.1**	ResNet	93.6	**CCP**	**70.1**

8. Learned Receptive Fields (CIFAR-10)



9. References

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

[2] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[3] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph Attention Networks. *ICLR*, 2018.