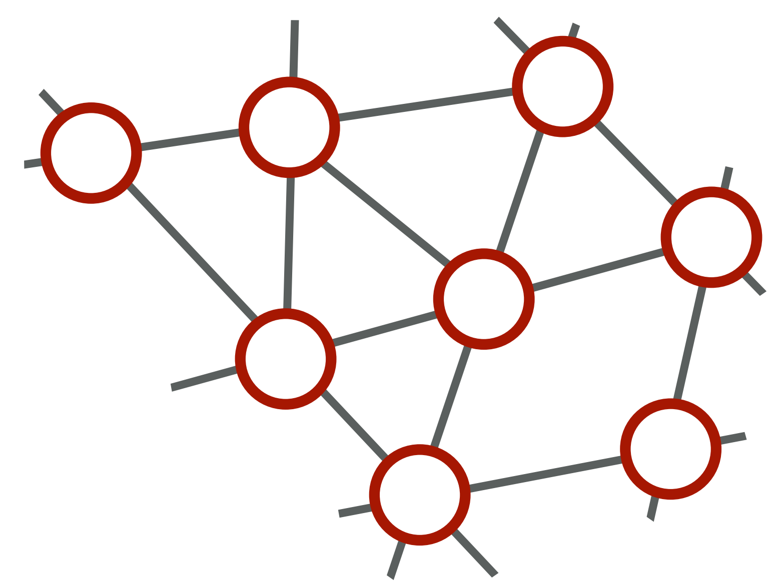
Machine Learning Laboratory

Course No. 389.247

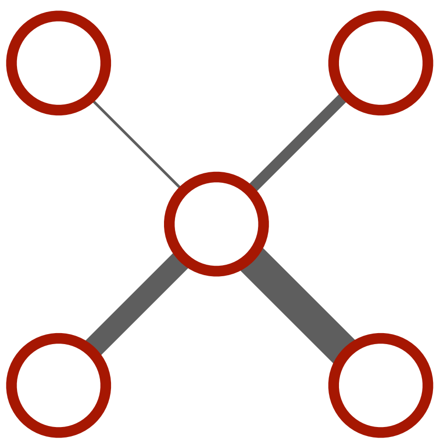
Graph Convolutional Networks

Gerald Matz & Dimitrios Kalodikis

institute of telecommunications



Graphs



$\mathcal{G} = (\mathcal{V}, \mathcal{E})$

$\mathcal{V} = \{1, \ldots, N\}$: nodes / vertices

$\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$: edges / lines (possibly weighted)

Nodes: users, devices, products, movies, brain regions, …

Edges: statistical relations, social relationships, communication links, biological pathways, …

Graph signal: $\mathbf{x}[n] : \mathcal{V} \to \mathbb{R}^L \quad \Rightarrow \mathbf{X} \in \mathbb{R}^{N \times L}$

ML perspective: node attributes, features

Algebraic Characterization

sparse

Weighted adjacency matrix: $\mathbf{W} \in \mathbb{R}^{N \times N}$

$(n,m) \in \mathcal{E} \; \ldots \; W_{nm}$ indicates strength of edge

$(n,m) \notin \mathcal{E} \; \ldots \; W_{nm} = 0$

Normalized graph Laplacian: $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$

with degree matrix $\mathbf{D} = \mathrm{Diag}(\mathbf{W1})$

Laplacian eigendecomposition:

$$\mathbf{L}\mathbf{u}_k = \lambda_k \mathbf{u}_k \quad \Longrightarrow \quad \mathbf{L} = \sum_k \lambda_k \mathbf{u}_k \mathbf{u}_k^{\mathrm{T}}$$

spectrum: $0 = \lambda_1 \leq \lambda_k \leq \lambda_N \leq 2$

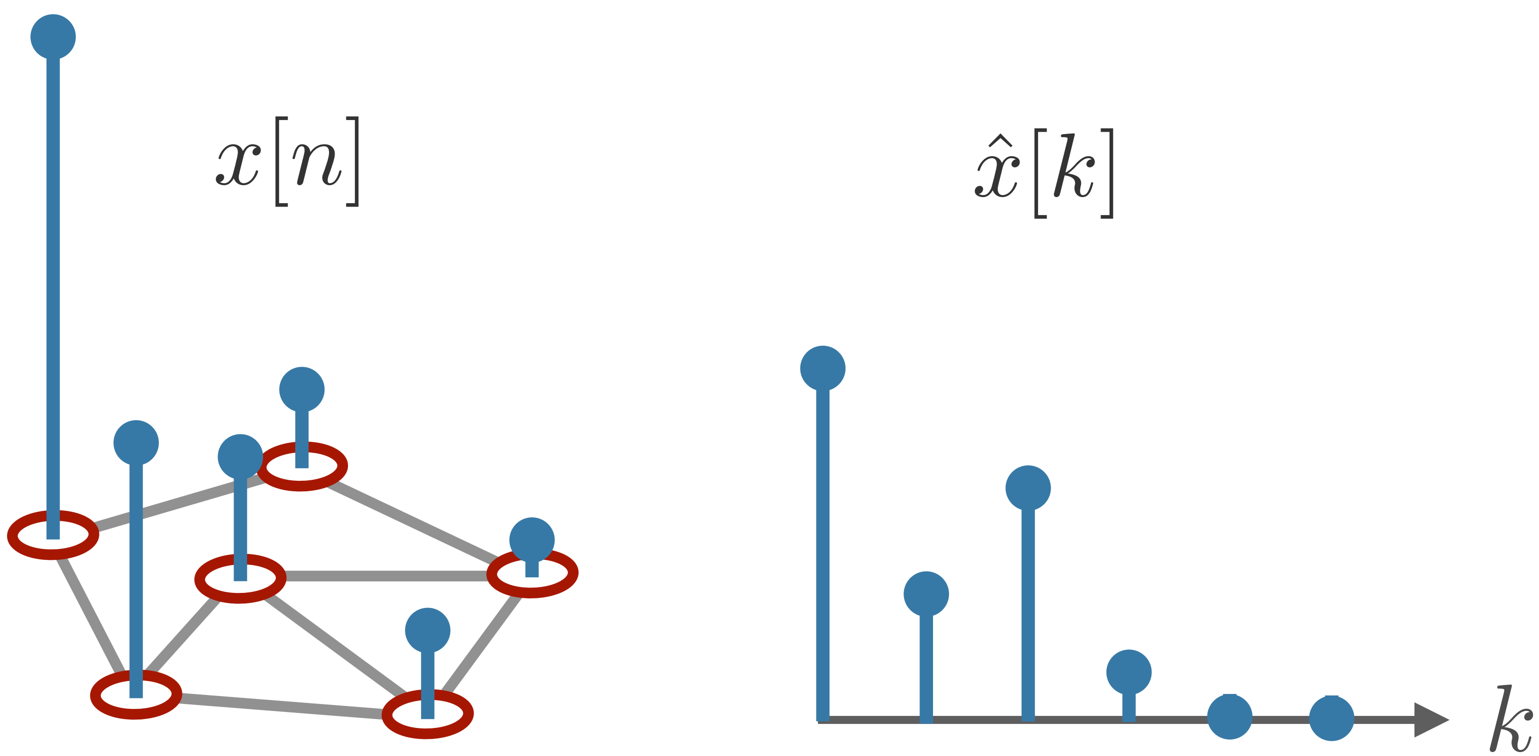
Graph Fourier Transform (GFT)

GFT: $\hat{x}[k] = \mathbf{u}_k^{\mathrm{T}}\mathbf{x}, \quad \mathbf{x} = \sum_{k=1}^{N} \hat{x}[k]\mathbf{u}_k$

not sparse

$$\hat{\mathbf{x}} = \mathbf{U}^{\mathrm{T}}\mathbf{x} \quad \Longleftrightarrow \quad \mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$$

Example:



Graph Filter ("Convolution")

I/O Relation: $\mathbf{y} = \mathbf{H}\mathbf{x}$ with $\mathbf{H} = p(\mathbf{L}) \triangleq \sum_k p(\lambda_k)\mathbf{u}_k\mathbf{u}_k^{\mathrm{T}}$

with polynomial $p(\lambda) = \sum_{i=0}^{I-1} \eta_i \lambda^i$

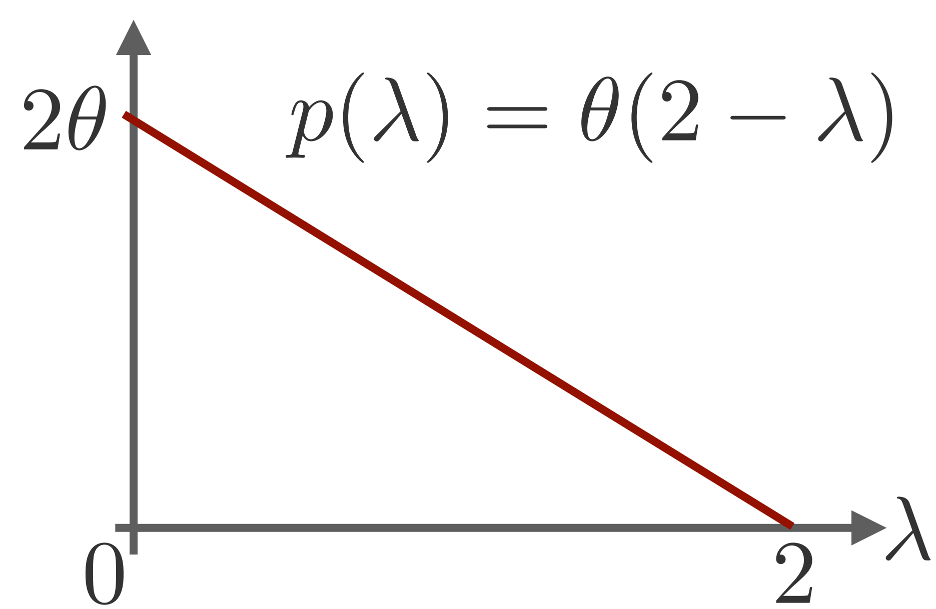
GFT Domain: $\hat{y}[k] = H[k]\hat{x}[k]$ with $H[k] = p(\lambda_k)$

Example: $I = 2,\ \eta_0 = 2\theta,\ \eta_1 = -\theta$

$$\mathbf{H} = 2\theta\mathbf{I} - \theta\mathbf{L}$$

$$= \theta(\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}})$$

$$y[n] = \theta x[n] + \theta \sum_{m \in \mathcal{N}(n)} \frac{W_{nm}}{\sqrt{d_n d_m}} x[m]$$



Graph Convolutional Network (GCN)

Basic concept:

- concatenate multiple layers
- each layer l consists of
  - a learnable graph convolution
  - an element-wise nonlinearity ("activation") $\sigma_l(\cdot)$

"Renormalization": replace $\mathbf{I} + \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$ with

$$\widehat{\mathbf{W}} = \widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{W}}\widetilde{\mathbf{D}}^{-\frac{1}{2}} \quad \text{where} \quad \widetilde{\mathbf{W}} = \mathbf{W} + \mathbf{I}$$

Multiple features: $$\mathbf{y}_i = \sum_j \mathbf{H}_{ij}\mathbf{x}_j = \sum_j \theta_{ji}\widehat{\mathbf{W}}\mathbf{x}_j$$

$$\Longrightarrow \quad \mathbf{Y} = \widehat{\mathbf{W}}\mathbf{X}\mathbf{\Theta}$$

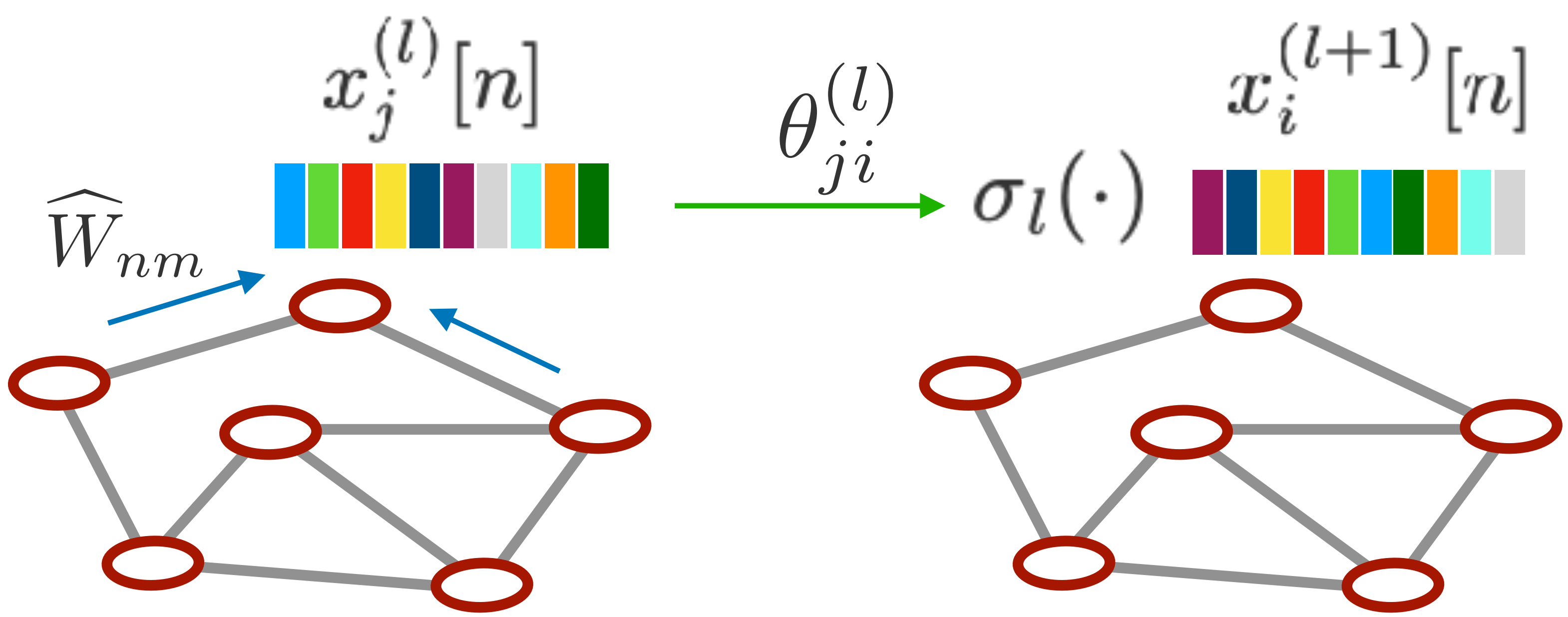
trainable parameters

Graph Convolutional Network (GCN)

Layer #*l*: $$\mathbf{X}^{(l+1)} = \sigma_l\big(\widehat{\mathbf{W}}\,\mathbf{X}^{(l)}\boldsymbol{\Theta}^{(l)}\big)$$

Elementwise reformulation:

$$x_i^{(l+1)}[n] = \sigma_l\left(\sum_{j=1}^{J}\theta_{ji}^{(l)}\sum_{m\in\mathcal{N}(n)\cup\{n\}}\widehat{W}_{nm}x_j^{(l)}[m]\right)$$



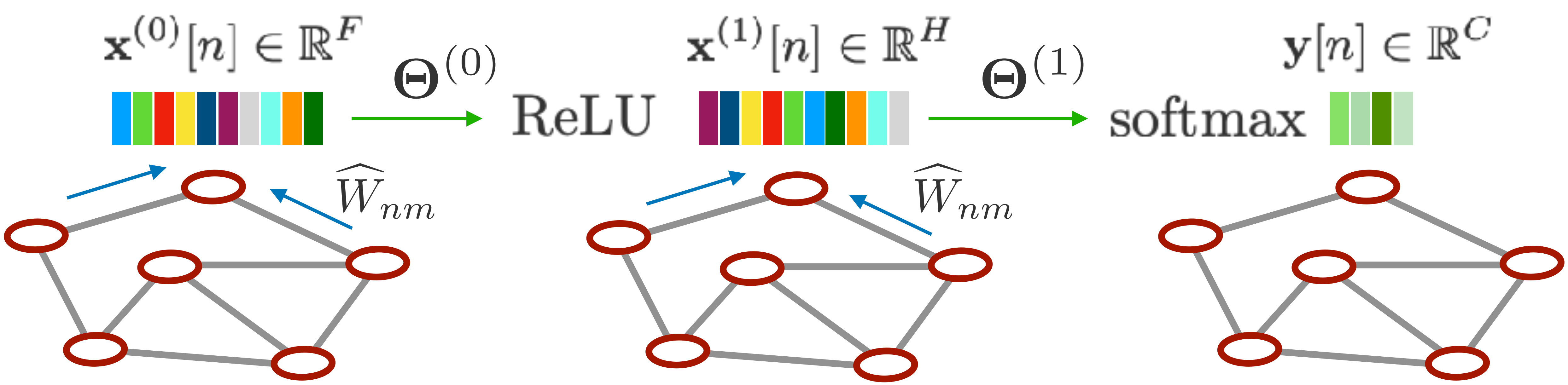
Semi-supervised Classification

Layer #1: $\mathbf{X}^{(1)} = \mathrm{ReLU}\left(\widehat{\mathbf{W}}\,\mathbf{X}^{(0)}\boldsymbol{\Theta}^{(0)}\right)$

Layer #2: $\mathbf{Y} = \mathbf{X}^{(2)} = \mathrm{softmax}\left(\widehat{\mathbf{W}}\,\mathbf{X}^{(1)}\boldsymbol{\Theta}^{(1)}\right)$

Training: - given P one-hot encoded node labels $p_j[n]$

- minimize cross-entropy $\mathcal{L} = -\sum_{n\in\mathcal{P}}\sum_{j=1}^{C} p_j[n]\log y_j[n]$



Transductive versus Inductive

Transductive learning

- works on training data and specific test data
- semi-supervised: some labels available
- no generalization to unseen test data
- in graph learning: typically just one graph

Inductive learning

- supervised training: all data labeled
- generalizes to unseen test data
- in graph learning: unseen test graphs