Spectral Graph Theory and Graph CNN

https://qdata.github.io/deep2Read

Presenter : Ji Gao

Outline

1. Graph Laplacian
   - Definitions
   - Why Laplacian?
   - Graph Fourier Transform
2. Spectral Neural Network
3. Fast Spectral Filtering
4. Reference

Outline

1. Graph Laplacian
   - Definitions
   - Why Laplacian?
   - Graph Fourier Transform
2. Spectral Neural Network
3. Fast Spectral Filtering
4. Reference

Graph

Graph

A graph $G = (V, E)$, where $V = 1, 2..N$ is the set of Vertices and $E \subseteq V \times V$.

(Vertex) Weighted Graph

A weighted graph $G = (V, E, W)$, where $V = 1, 2..N$ is the set of Vertices, $E \subseteq V \times V$, $W : V \rightarrow R$.

Graph

Degree

The degree $d(v)$ of a vertex v is the number of vertices in G that are adjacent to v.

Adjacency Matrix

Adjacency matrix A of the graph G is a $n \times n$ matrix that

$$A_{ij} = \begin{cases} 1 & (i,j) \in E \\ 0 & \text{Otherwise} \end{cases}$$

Graph Laplacian

(Unnormalized) Graph Laplacian

Graph Laplacian $L = \text{diag}(d) - A$, which

$$L_{ij} = \begin{cases} d_i & i = j \\ -1 & i \neq j \& (i,j) \in E \\ 0 & \text{Otherwise} \end{cases}$$

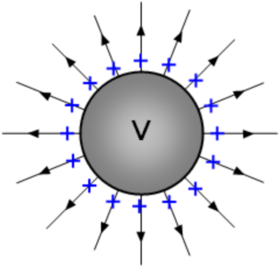
Outline

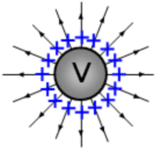
Why Laplacian?

Laplacian

For function f, Laplacian operator $\Delta f = \nabla \cdot \nabla f$

- Laplacian represents the divergence of the gradient.
- It's a coordinate-free operator!
- In physics, if a electromagnetic field is defined by a electrostatic potential function ϕ, then $\Delta\phi$ gives the charge distribution in the field.





Eigenfunction of Laplacian Operator

Eigenfunction of Laplacian in $(0, 1)$

Suppose f is the eigenfunction of the Laplacian:

$$\Delta f + \lambda f = 0, f(0) = f(1) = 0$$

$$\Delta f = \frac{\partial^2 f}{\partial x^2} = -\lambda f$$

The only non-trivial solution of the Laplacian is

$$f_n(x) = C \sin(n\pi x), n \in N$$

- f_n is the Fourier sine series.
- f_n together forms an orthonormal basis of the space $L^2(0, 1)$
- Theorem: For any $L^2(\Omega)$ space where Ω is a reasonably smooth domain, there exists an orthonormal family of eigenfunctions of Δ that forms an orthonormal basis of the space.

Outline

1. Graph Laplacian
   - Definitions
   - Why Laplacian?
   - Graph Fourier Transform
2. Spectral Neural Network
3. Fast Spectral Filtering
4. Reference

Graph Laplacian Revisited

Graph Laplacian

Graph Laplacian $L = D - A$

- Suppose f is a function from vertex to $\mathcal{R}$.
- f can be represented by a vector $(f_1, f_2...f_n)$ with size n.
- Therefore, $[Lf]_i = d_i - \sum_j A_{ij} f_j = \sum_j A_{ij}(f_i - f_j)$
- Calculating the difference on the value of a vertex to its neighbors!
- $$f^T L f = \sum_{<i,j>\in E} (f_i - f_j)^2$$

Graph Laplacian Revisited

Graph Laplacian

$$L = D - A$$

$$f^T L f = \sum_{<i,j>\in E} (f_i - f_j)^2$$

- Symmetric real matrix $\longrightarrow$ Real eigenvalues
- Positive semidefinite $\longrightarrow$ Non-negative eigenvalues
- First eigenvalue is 0 with eigenvector $\{1, 1, 1...1\}$
- $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$

Graph Fourier transform

- The eigenvector of graph Laplacian matrix can be used as a orthonormal basis of the Hilbert space.

one-dimensional Laplace operator: $\frac{d^2}{dx^2}$

eigenfunctions: $e^{j\omega x}$

Classical FT: $\hat{f}(\omega) = \boxed{\int}\boxed{(e^{j\omega x})^*}\boxed{f(x)}dx$

$$f(x) = \frac{1}{2\pi}\int \hat{f}(\omega)e^{j\omega x}d\omega$$

graph Laplacian: L

eigenvectors: χ_ℓ

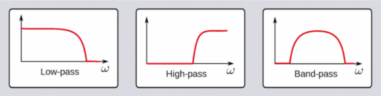
$f : V \to \mathbb{R}^N$

Graph FT: $\hat{f}(\ell) = \langle \chi_\ell, f \rangle = \boxed{\sum_{i=1}^{N}}\boxed{\chi_\ell^*(i)}\boxed{f(i)}$

$$f(i) = \sum_{\ell=0}^{N-1} \hat{f}(\ell)\chi_\ell(i)$$

Graph Spectral Filtering

$$f \xrightarrow{\text{FT}} \hat{f}(\omega) \xrightarrow{\hat{g}(\omega)} \hat{g}(\omega)\hat{f}(\omega) \xrightarrow{\text{IFT}} f * g$$



$$f \xrightarrow{\text{GFT}} \chi^T f \xrightarrow{\hat{g}(\Lambda)} \hat{g}(\Lambda)\chi^T f \xrightarrow{\text{IGFT}} \chi\hat{g}(\Lambda)\chi^T f$$

$$\hat{g}(\Lambda) = \begin{bmatrix} \hat{g}(\lambda_0) & & 0 \\ & \ddots & \\ 0 & & \hat{g}(\lambda_{N-1}) \end{bmatrix}$$

- Filters can be used to form a convolutional layer

Spectral Networks and Deep Locally Connected Networks on graphs

Joan Bruna, Wojciech Zaremba, Arthur Szlam, Yann Lecun

- CNN is powerful. Extend CNN to general graphs.
- 1. Use **hierarchical clustering**
- 2. Use **spectrum of graph laplacian** to learn convolutional layers
- **Efficient**: Number of parameters is independent of input size

Spatial CNN use Hierarchical clustering

- Form a multi-scale clustering
- The k-th layer has d_k clusters
- The k-th layer has f_k filters

Convolutional Layer

For $j = 1..f_k$,

$$x_{k+1,j} = L_k h(\sum_{i=1}^{f_{k-1}} F_{k,i,j} x_{k,i})$$

.

F_k, i, j is a $d_{k-1} \times d_{k-1}$ sparse matrix.
L_k is a pooling operation.

- Clusters are pre-defined by hierarchical clustering.

Spectral Construction

Spectral Convolution

Suppose V is the eigenvectors of L.
Input: x_k, size $n \times f_{k-1}$
Without spatial subsampling:

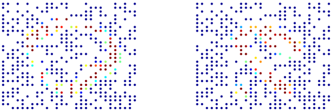
$$x_{k+1,j} = h(U \sum_{i=1}^{f_{k-1}} F_{k,i,j} U^T x_{k,i})$$

$F_{k,i,j}$ is a diagonal weight matrix.

- Only use top d eigenvectors to reduce cost.

Experiment 1: Subsampled MNIST

- Subsample MNIST to 400 points
- Baseline: Nearest Neighbor (4.11% Error rate)

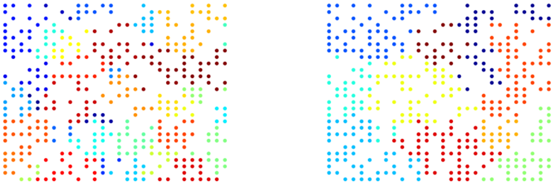


(a) (b)

Figure 3: Subsampled MNIST examples.

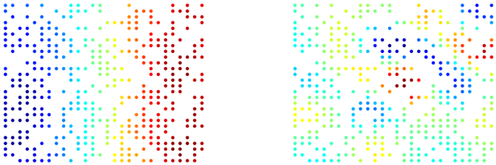
Table 1: Classification results on MNIST subsampled on 400 random locations, for different architectures. FCN stands for a fully connected layer with N outputs, LRFN denotes the locally connected construction from Section 2.3 with N outputs, MPN is a max-pooling layer with N outputs, and SPN stands for the spectral layer from Section 3.2.

method	Parameters	Error
Nearest Neighbors	N/A	4.11
400-FC800-FC50-10	$3.6 \cdot 10^5$	1.8
400-LRF1600-MP800-10	$7.2 \cdot 10^4$	1.8
400-LRF3200-MP800-LRF800-MP400-10	$1.6 \cdot 10^5$	**1.3**
400-SP1600-10 ($d_1 = 300$, $q = n$)	$3.2 \cdot 10^3$	2.6
400-SP1600-10 ($d_1 = 300$, $q = 32$)	$1.6 \cdot 10^3$	2.3
400-SP4800-10 ($d_1 = 300$, $q = 20$)	$5 \cdot 10^3$	1.8



(a) (b)

Figure 4: Clusters obtained with the agglomerative clustering. (a) Clusters corresponding to the finest scale $k = 1$, (b) clusters for $k = 3$.

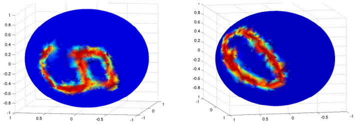


(a) (b)

Figure 5: Examples of Eigenfunctions of the Graph Laplacian v_2, v_{20}.

Experiment 2: Sphere MNIST

- Project MNIST to sphere
- Uniformly or Randomly



(a) (b)

Figure 7: Examples of some MNIST digits on the sphere.

Table 2: Classification results on the MNIST-sphere dataset generated using partial rotations, for different architectures

method	Parameters	Error
Nearest Neighbors	N/A	19
4096-FC2048-FC512-9	10^7	**5.6**
4096-LRF4620-MP2000-FC300-9	$8 \cdot 10^5$	**6**
4096-LRF4620-MP2000-LRF500-MP250-9	$2 \cdot 10^5$	6.5
4096-SP32K-MP3000-FC300-9 ($d_1 = 2048$, $q = n$)	$9 \cdot 10^5$	7
4096-SP32K-MP3000-FC300-9 ($d_1 = 2048$, $q = 64$)	$9 \cdot 10^5$	**6**

Experiment 2: Sphere MNIST

Table 3: Classification results on the MNIST-sphere dataset generated using uniformly random rotations, for different architectures

method	Parameters	Error
Nearest Neighbors	NA	80
4096-FC2048-FC512-9	10^7	52
4096-LRF4620-MP2000-FC300-9	$8 \cdot 10^5$	61
4096-LRF4620-MP2000-LRF500-MP250-9	$2 \cdot 10^5$	63
4096-SP32K-MP3000-FC300-9 ($d_1 = 2048, q = n$)	$9 \cdot 10^5$	56
4096-SP32K-MP3000-FC300-9 ($d_1 = 2048, q = 64$)	$9 \cdot 10^5$	**50**

- Improve previous spectral CNN
- Main Contributions:
  - Strictly localized filters
  - Low computational complexity
  - Efficient pooling method
  - Multiple experiment on different datatypes

Filters

graph filter

$$y = U^T g(\Lambda) U x$$

Where U is the eigenvector of L and Λ is the diagonal matrix of all eigenvalues of L

- Naive approach is to learn $g(\Lambda) = diag(\theta)$ directly.
- Limitations:
    - It's not localized
    - The complexity is O(n).

Polynomial filter

Polynomial filter

$g(\Lambda) = \sum_{k=1}^{L} \theta_k \Lambda^K$

- Spectral filters represented by Kth-order polynomials of the Laplacian are K-localized: connect all the vertices in at most K steps.
- Learning complicity is $O(K)$
- Use Chebyshev polynomial to make it faster: $g(\Lambda) = \sum_{k=1}^{L} \theta_k T_k(\Lambda)$, where $T_k = 2xT_{k-1} - T_{k-2}$

Pooling

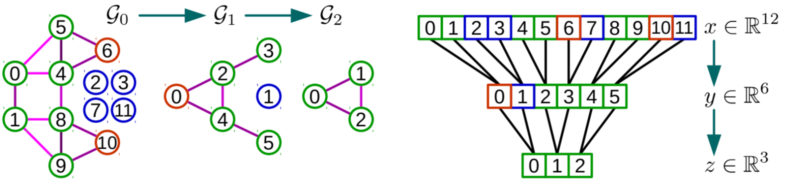


Figure 2: **Example of Graph Coarsening and Pooling.** Let us carry out a max pooling of size 4 (or two poolings of size 2) on a signal $x \in \mathbb{R}^8$ living on $\mathcal{G}_0$, the finest graph given as input. Note that it originally possesses $n_0 = |\mathcal{V}_0| = 8$ vertices, arbitrarily ordered. For a pooling of size 4, two coarsenings of size 2 are needed: let Graclus gives $\mathcal{G}_1$ of size $n_1 = |\mathcal{V}_1| = 5$, then $\mathcal{G}_2$ of size $n_2 = |\mathcal{V}_2| = 3$, the coarsest graph. Sizes are thus set to $n_2 = 3$, $n_1 = 6$, $n_0 = 12$ and fake nodes (in blue) are added to $\mathcal{V}_1$ (1 node) and $\mathcal{V}_0$ (4 nodes) to pair with the singeltons (in orange), such that each node has exactly two children. Nodes in $\mathcal{V}_2$ are then arbitrarily ordered and nodes in $\mathcal{V}_1$ and $\mathcal{V}_0$ are ordered consequently. At that point the arrangement of vertices in $\mathcal{V}_0$ permits a regular 1D pooling on $x \in \mathbb{R}^{12}$ such that $z = [\max(x_0, x_1), \max(x_4, x_5, x_6), \max(x_8, x_9, x_{10})] \in \mathbb{R}^3$, where the signal components x_2, x_3, x_7, x_{11} are set to a neutral value.

Experiment 1: MNIST

Model	Architecture	Accuracy
Classical CNN	C32-P4-C64-P4-FC512	99.33
Proposed graph CNN	GC32-P4-GC64-P4-FC512	99.14

Table 1: Classification accuracies of the proposed graph CNN and a classical CNN on MNIST.

Experiment 2: 20Newsgroup

Model	Accuracy
Linear SVM	65.90
Multinomial Naive Bayes	68.51
Softmax	66.28
FC2500	64.64
FC2500-FC500	65.76
GC32	68.26

Table 2: Accuracies of the proposed graph CNN and other methods on 20NEWS.

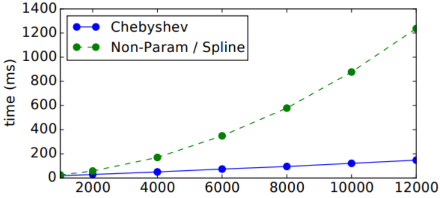


Figure 3: Time to process a mini-batch of $S = 100$ 20NEWS documents w.r.t. the number of words n.

Reference

1. Laplacian Operator - Wikipedia
2. An introduction to spectral graph theory *Jiang Jiaqi*
3. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering *Michaël Defferrard, Xavier Bresson, Pierre Vandergheynst(EPFL, Lausanne, Switzerland)*
4. Spectral Networks and Locally Connected Networks on Graphs *Joan Bruna, Wojciech Zaremba, Arthur Szlam, Yann LeCun*
5. Graph signal processing: Concepts, tools and applications *Xiaowen Dong*