

Notes on Geometric Learning

Matthew Mo

2020 年 7 月 23 日

目录

1	Math Backgrounds	1
1.1	Analysis on \mathbb{R}^n	1
1.2	Differential Manifold	1
1.3	Calculus on Manifold	1
1.4	Graph and Discrete Differential Operators	2
1.5	Fourier Analysis on Manifolds	3
2	Vanilla GNN(Scarselli et al.)	3
2.1	Model	4
2.2	Limitations	4
3	Spectral Methods	5
3.1	Spectral CNN(SCNN, Bruna et al.)	5
3.2	Graph Coarsening	5
3.3	Spectral CNN + 光滑谱滤波器	6
4	Non-Spectral Methods	6
4.1	Graph CNN(GCNN a.k.a. ChebNet, Defferd et al.)	6
4.2	Graph Convolutional Net(GCN, Kipf, Welling et al.)	7
4.3	Adaptive GCN/AGCN, [Li et al., 2018b]	8
4.4	Graph Neural Net(GNN)	8
5	Charting-based Methods	8
5.1	Geodesic CNN(测地 CNN)	9
5.2	Anisotropic CNN(各向异性 CNN)	9
5.3	Dynamic Graph CNN(DGCNN)	9
5.4	Mixture Model Network(MoNet, Monti et al.)	10

6 Spatial Methods	11
6.1 Neural FPS (Duvenaud et al., 2015)	11
6.2 PATCHY-SAN (Niepert et al., 2016)	11
6.3 Diffusion-Convolutional NN/DCNN (Atwood and Towsley, 2016)	11
6.4 Dual GCN/DGCN (Zhuang and Ma, 2018)	11
6.5 Learnable GCN/LGCN (Gao et al., 2018)	12
7 Spatial-Spectral Methods	12
7.1 Windowed Fourier Transform	12
7.2 Wavelet Transform	13
7.3 Localized Spectral CNN (LSCNN, Boscaini et al.)	13
8 Application	13
9 Open Problems and Discussions	14

1 Math Backgrounds

1.1 Analysis on \mathbb{R}^n

1.2 Differential Manifold

一个微分流形 (differential manifold) 基本上就是一个局部欧式空间的拓扑空间. 严格上来说: 一个 d 维微分流形 \mathcal{X} 是一个拓扑空间, 并且对于每一个其中的点都存在一个邻域与 d 维欧氏空间 (i.e. \mathbb{R}^d) 同胚 (homeomorphic), 称作切空间 (tangent space) $T_x\mathcal{X}$, 所有切空间一起 (的并) 组成了切丛 (tangent bundle) $T\mathcal{X}$, 并且可以定义内积 $\langle \cdot, \cdot \rangle_{T_x\mathcal{X}} : T_x\mathcal{X} \times T_x\mathcal{X} \rightarrow \mathbb{R}$, 关于 x 是光滑的, 成为流形上的一个黎曼度量 (Riemann metric). 后文中的流形若非特殊指出意味着微分流形. 一个流形和其上的黎曼度量称为黎曼流形. 注意, 以上定义中的任何定义都是内蕴的, 图也可以是一个流形!

Target 将在欧氏空间上大获成功的 CNN 方法推广到流形 (以及图) 上!

Theorem (Whitney weak embedding theorem) 任意 n 维微分流形 M 可以光滑地嵌入 \mathbb{R}^{2n+1} , 以及光滑的浸入 \mathbb{R}^{2n} . 并且每个连续映射 $M \rightarrow \mathbb{R}^{2n+1}$ ($M \rightarrow \mathbb{R}^{2n+1}$) 可以用光滑嵌入 (浸入) 逼近, 也即这些光滑嵌入 (在所有连续映射中) 是稠密的.

一个更强的定理指出, 任意 n 维微分流形 M 可以光滑地嵌入 \mathbb{R}^{2n} , 但它们不再稠密.

Theorem (Nash embedding theorem) 任意 Riemann 流形可以保距地光滑嵌入某个维度的欧氏空间中.

1.3 Calculus on Manifold

我们关心两种流形上的函数: 标量场 $f : M \rightarrow \mathbb{R}$, 切向量场 $F : M \rightarrow TM$. 可以定义流形上函数的内积, 这使平方可积函数形成 Hilbert 空间:

$$\langle f, g \rangle_{L^2(M)} = \int_M f(x)g(x)dx \quad (1)$$

$$\langle F, G \rangle_{L^2(TM)} = \int_M \langle F(x)G(x) \rangle_{L^2(M)} dx \quad (2)$$

其中 dx 是 Riemann 度量诱导的体积微元.

f 的微分是一个算子: $df : TM \rightarrow \mathbb{R}$ 作用在切向量场上, 在 $x \in M$ 上, 微分可以被定义为一个 1-形式:

$$df(x) = \langle \nabla f(x), \cdot \rangle_{T_x M} \quad (3)$$

以及导数

$$df(x)F(x) = \langle \nabla f(x), F(x) \rangle_{T_x M} \quad (4)$$

这是一种方向导数的推广. 有内蕴导数 (梯度)

$$\nabla f : L^2(M) \rightarrow L^2(TM) \quad (5)$$

对称地有它的形式伴随算子内蕴散度

$$\text{div} f : L^2(TM) \rightarrow L^2(M) \quad (6)$$

因为

$$\langle F(x), \nabla f(x) \rangle_{L^2(TM)} = \langle -\text{div} F(x), f(x) \rangle_{L^2(M)}$$

Laplace-Beltrami 算子: $\Delta f = -\text{div} \nabla f$. 它是自伴 (对称) 算子.

Dirichlet 能量: $\|\nabla f\|_{L^2(TM)}^2 = \langle \nabla f, \nabla f \rangle_{L^2(TM)} = \langle f, \Delta f \rangle_{L^2(M)}$ 衡量了一个函数的光滑程度

注意: 以上定义无关流形地外含性质, 是内蕴地定义. 于是我们可以在切空间上定义一个基, 并且用 d -向量表示切向量, 以及 $d \times d$ 对称矩阵表示 Riemann 度规.

1.4 Graph and Discrete Differential Operators

考虑带权无向图 $(\mathcal{V}, \mathcal{E})$, 以及边和点权

$$\begin{cases} w_{ij} \geq 0, \forall (i, j) \in \mathcal{E} \\ a_i > 0, \forall i \in \mathcal{V} \end{cases} \quad (7)$$

定义其上的内积并给出 Hilbert 空间:

$$\begin{cases} \langle f, g \rangle_{L^2(\mathcal{V})} = \sum_{v \in \mathcal{V}} a_v f_v g_v \\ \langle F, G \rangle_{L^2(\mathcal{E})} = \sum_{e \in \mathcal{E}} w_{ij} F_{ij} G_{ij} \end{cases} \quad (8)$$

图梯度 $\nabla : L^2(\mathcal{V}) \rightarrow L^2(\mathcal{E})$:

$$\nabla f_{ij} = f_i - f_j \quad (9)$$

图梯度 $\nabla : L^2(\mathcal{E}) \rightarrow L^2(\mathcal{V})$:

$$\text{div} F_i = \frac{1}{a_i} \sum_{(i,j) \in \mathcal{E}} w_{ij} F_{ij} \quad (10)$$

图 Laplace 算子 $\Delta : L^2(\mathcal{V}) \rightarrow L^2(\mathcal{V})$:

$$\Delta f = -\text{div} \nabla f \Delta f_i = \frac{1}{a_i} \sum_{(i,j) \in \mathcal{E}} w_{ij} (f_i - f_j) \quad (11)$$

下设 $\mathbf{W} = (w_{ij})$, $\mathbf{A} = \text{Diag}(a_i)$, $\mathbf{f} = (f_i)^T$, $\mathbf{D} = \text{Diag}(\sum_{j,j \neq i} w_{ij})$ 则

$$\Delta f = \mathbf{A}^{-1}(\mathbf{D} - \mathbf{W})\mathbf{f} \quad (12)$$

$\mathbf{A} = \mathbf{I} \rightarrow$ unnormalized graph Laplacian.

$\mathbf{A} = \mathbf{D} \rightarrow$ randomwalk graph Laplacian.

1.5 Fourier Analysis on Manifolds

Δ 是自伴, 半正定算子, 在紧域上有正交特征值分解, 记它们是 ϕ_0, \dots , 并且其中有 $\langle \phi_i, \phi_j \rangle_{L^2(M)} = \delta_{ij}$ 和对应非负特征值 $0 \leq \lambda_0 \leq \lambda_1 \leq \dots$ (Δ 的谱).

Idea ϕ_i 是黎曼流形上最光滑的函数 (在 Dirichlet 能量意义上), 可以作为 Fourier 分析 (Hilbert 空间) 的基!

计算 $f \in L^2(M)$ 的展开:

$$f(x) = \sum_{i \geq 0} \langle f, \phi_i(x) \rangle_{L^2(M)} \phi_i(x) \quad (13)$$

在欧式域上的卷积有性质:

$$\widehat{(f * g)}(\omega) = \hat{f}(\omega) \hat{g}(\omega) \quad (14)$$

但是在非欧域上不能简单地定义卷积 ($\tau - x$ 怎么定义?), 那么可以由卷积性质定义卷积:

$$(f * g)(x) = \sum_{i \geq 0} \langle f, \phi_i \rangle_{L^2(M)} \langle g, \phi_i \rangle_{L^2(M)} \phi_i(x) \quad (15)$$

但缺乏平移不变性!

在图上, Laplace 算子有 n 个特征值向量, 并且

$$f * g \Rightarrow \mathbf{G}f = \Phi \text{diag}(\hat{\mathbf{g}}) \Phi^T f \quad (16)$$

其中 $\hat{g} = (\hat{g}_i)$ 是谱表示, $\Phi = (\phi_i)$ 是 Laplace 算子的特征矩阵.

Note 卷积和 Laplace 算子交换 $\Delta \mathbf{G}f = \mathbf{G} \Delta f$.

Note Laplace 算子的特征向量组可能不是唯一的 (比如符号取反即可), 同构域也可能有不同的基, 这代表着域结构的小扰动会导致特征向量组的巨大变化, 这会导致基于它们的算法难以跨域. 有些 kernel 似乎没有这些问题 (heat kernel, diffusion kernel).

2 Vanilla GNN(Scarselli et al.)

Target node $v \Rightarrow$ representation $\mathbf{h}_v \in \mathbb{R}^s \Rightarrow$ out \mathbf{o}_v

Domain undirected graph + node feature \mathbf{x}_v + possible edge feature.

$co[v], ne[v]$: edges, neighbors of v .

2.1 Model

$$h_v = f(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]}) \quad (17)$$

$$o_v = g(h_v, x_v) \quad (18)$$

where f : local transition function, g : local out function, both parametric. Let $\mathbf{H}, \mathbf{O}, \mathbf{X}, \mathbf{X}_N$ are those states in mat form

$$\mathbf{H} = F(\mathbf{H}, \mathbf{X}) \quad (19)$$

$$\mathbf{O} = F(\mathbf{H}, \mathbf{X}_N) \quad (20)$$

F : global transition function, G : global out function, both parametric, use fixed point iteration to calculate \mathbf{H} :

$$\mathbf{H}^{t+1} = F(\mathbf{H}^t, \mathbf{X}) \quad (21)$$

write loss using supervisory target info t_i

$$loss = \sum_{i=1}^p (t_i - o_i) \quad (22)$$

p is node count supervised.

Note An Implicit Neuron Representation!

Algorithm

1. update h'_v from (17) for T time steps
2. compute loss and gradients
3. update parameters

Note equals T layers of GNN of same parameters

2.2 Limitations

- computational inefficient
- each layer share same parameters \Rightarrow goto either RNN(GRU/LSTM)/ use different params

- edge features?
- large $T \Rightarrow$ graph representation be smooth, hard to distinguish nodes
- **Further Nets**
 - GGNN(computational efficiency)
 - R-GCN(directed graph)

3 Spectral Methods

3.1 Spectral CNN(SCNN, Bruna et al.)

一个谱卷积层:

$$g_l = \xi \left(\sum_{l'=1}^p \Phi_k \Gamma_{l,l'} \Phi_k^T f_{l'} \right) \quad (23)$$

其中:

- ξ 是某种非线性函数, 如 *ReLU, tanh, sigmoid*
- $F = (\mathbf{f}_1, \mathbf{f}_q) \in K^{n \times q}, G = (\mathbf{g}_1, \dots, \mathbf{g}_p) \in K^{n \times p}$ 是 p/q 个 I/O 信号 (在图的顶点上, $n = |\mathcal{V}|$)
- $\Gamma_{l,l'} \in K^{k \times k}$ 是一个对角阵, 代表了频域的一个滤波器
- Φ_k 是前 k 个特征向量组成的矩阵, 代表了只用了前 k 个特征向量的截止频率, 一般应用中 $k \ll n$
- $\Delta = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, 利用的是 normalized Laplace 算子.

这种方法:

- 仅限于单一域: $\Gamma_{l,l'}$ 取决于基的选取.
- 可以在跨域上建立正交基, 但需要域的先验知识, 如域之间的对应, 然而后者常常是难以找到的
- 设 $k = O(n)$ 个特征向量被保留, 则一个卷积层中有 $pqk = O(n)$ 个参数, 如果让 $k = O(1/n)$ 则可以使之类似于 CNN, 参数数和输入大小无关!

3.2 Graph Coarsening

图粗化就是池化的类比: 以 $\alpha < 1$ 的比例保持图顶点, 那么令 $\Phi, \tilde{\Phi}$ 是 $n \times n, \alpha n \times \alpha n$ 的特征值矩阵, 则有

$$\tilde{\Phi} \approx P \Phi \begin{pmatrix} I_{\alpha n} \\ 0 \end{pmatrix} \quad (24)$$

其中 $P \in \{0,1\}^{an \times n}$, 其中 i 行编码了第 i 顶点在粗化图中的位置. 然而由于应用了非线性, 要在粗化后每层计算特征矩阵.

Note 单个高频分量 \Rightarrow 不稳定, 把高频分量组合 \Rightarrow 有意义的信息.

3.3 Spectral CNN + 光滑谱滤波器

为了降低 overfit, 需要降低参数个数: 亟需限制谱乘子的类别 $(\Gamma_{l,l'})$, 在频域上表达局域性. 在欧式空间上有 Pascal 恒等式:

$$\int_{\mathbb{R}} |x|^{2k} |f(x)|^2 dx = \int_{\mathbb{R}} \left| \frac{\partial^k \tilde{f}(\omega)}{\partial \omega^k} \right| d\omega \quad (25)$$

Idea

- 学一个时域局域化 filter \Rightarrow 学一个光滑 filter
- 在一个频率的子集上学习谱乘子, 再内插 (如三次样条)!

有

$$diag(\Gamma_{l,l'}) = \mathbf{B} \boldsymbol{\alpha}_{l,l'} \gamma_i = \sum_{j=1}^q \alpha_j \beta_j(\lambda_i) \quad (26)$$

其中 $\mathbf{B} = (b_{ij}) = (\beta_j(\lambda_i)) \in \mathbb{R}^{k \times q}$ 是固定的内插 kernel (如 $\beta_j(\lambda)$ 是三次样条), $\boldsymbol{\alpha}$ 是 q 个内插系数.

若选择 $n\gamma^{-1} = O(1)$ 个内插系数, 使参数与 n 无关, 总模型有 $O(\log n)$ 个可训练参数 (why?). 但是 BP/FP 的复杂度仍然很高, 这是由于 Φ_k 和 Φ_k^T 的乘法复杂度很高, 在欧式域上有 FFT 算法, 是 $O(n \log n)$ 的, 但是在普通意义上的图上没有此类算法, 只能 $O(n^2)$. \Rightarrow 接下来会看到避免此计算的方法.

4 Non-Spectral Methods

Idea Laplace 算子的多项式在特征值上相同地作用, 故可以用多项式展开代替在特征值上的作用:

$$g_{\alpha}(\Delta) = \Phi g_{\alpha}(\Lambda) \Phi^T, \quad (27)$$

$$g_{\alpha}(\lambda) = \sum_{j=0}^{r-1} \alpha_j \lambda^j \quad (28)$$

$$\Lambda = diag(\lambda_i) \quad (29)$$

此时 $\Gamma_{l,l'} = g_{\alpha_{l,l'}}(\Lambda)$, 自动生成局域化 filter. 由于 Δ^r 是 r -局域化的, 所以上面的 filter 就像是作用在 r -hops 上的 diffusion kernel.

4.1 Graph CNN(GCNN a.k.a. ChebNet, Defferd et al.)

使用 Chebyshev 多项式:

$$T_0(\lambda) = 1, T_1(\lambda) = \lambda, \quad (30)$$

$$T_j(\lambda) = 2\lambda T_{j-1}(\lambda) - T_{j-2}(\lambda) \quad (31)$$

$$T_n(x) = \cos(n \arccos(x)) \quad (32)$$

展开 filter:

$$g_\alpha(\tilde{\Delta}) = \sum_{j=0}^{r-1} \alpha_j \Phi T_j(\tilde{\Lambda}) \Phi^T = \sum_{j=0}^{r-1} \alpha_j T_j(\tilde{\Delta}) \quad (33)$$

其中 $\tilde{\Delta} = 2\lambda_n^{-1}\Delta - \mathbf{I}$, $\tilde{\Lambda} = 2\lambda_n^{-1}\Lambda - \mathbf{I}$, 代表了 $[0, \lambda_n] \Rightarrow [-1, 1]$ 的归一化.

记 $\bar{f}^{(j)} = T_j(\tilde{\Delta})f$, 可计算

$$\bar{f}^{(j)} = 2\tilde{\Delta}\bar{f}^{(j-1)} - \bar{f}^{(j-2)} \quad (34)$$

$$\bar{f}^{(1)} = \tilde{\Delta}\bar{f} \quad (35)$$

$$\bar{f}^{(0)} = f \quad (36)$$

这是 $O(rn)$ 的, 并且无需显式计算 Laplace 算子. 以及总 filter:

$$g_\alpha(\tilde{\Delta})f = \sum_{j=0}^{r-1} \alpha_j T_j(\tilde{\Delta})f \quad (37)$$

$$= \sum_{j=0}^{r-1} \alpha_j \bar{f}^{(j)} \quad (38)$$

每层:

$$y_{s,i} = \sum_j g_{\alpha_{i,j}}(\tilde{\Delta})x_{s_j} \quad (39)$$

采用的图粗化技术 (Graclus): 在每一级图粗化水平上, 我们遍历结点, 当 i 结点遍历到时, 找一个邻接点使得局部割 $W_{ij}(1/d_i + 1/d_j)$ 最小化; 两个匹配结点被合并为一个, 标记已遍历并权值相加. 实行这种池化操作时, 会得到一个平衡二叉树, 这里, 若一个结点只有一个子节点, 那么给它增加一个 fake node, 并初始化其上的信号为 0, 且不与其它节点相连.

4.2 Graph Convolutional Net(GCN, Kipf, Welling et al.)

Idea 简化 GCNN 的设定: $r = 2, \lambda_n \approx 2$, 这给出

$$g_\alpha(f) = \alpha_0 f + \alpha_1(\Delta - \mathbf{I})f = \alpha_0 f + \alpha_1 \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2} f \quad (40)$$

Idea 更进一步 $\alpha = \alpha_0 = -\alpha_1$, 这给出

$$g_\alpha(f) = \alpha(\mathbf{I} + \mathbf{D}^{-1/2} \mathbf{W} \mathbf{D}^{-1/2})f \quad (41)$$

又由于 $(\mathbf{I} + \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2})$ 的元素取值于 $[0, 2]$, 会导致数值不稳定, 所以重整化之:

$$g_\alpha(f) = \alpha \widetilde{\mathbf{D}}^{-1/2} \widetilde{\mathbf{W}} \widetilde{\mathbf{D}}^{-1/2} f \quad (42)$$

其中 $\widetilde{\mathbf{W}} = \mathbf{W} + \mathbf{I}$, $\widetilde{\mathbf{D}} = \text{diag}(\sum_{j \neq i} \tilde{w}_{ij})$, TC.: $O(|\mathcal{E}|FC)$

为了解决半监督图学习问题, 可以通过带正则 Laplace 算子项的损失函数来训练:

$$L = L_0 + \lambda L_{reg}, L_{reg} = \sum_{i,j} A_{ij} \|f(X_i) - f(X_j)\|^2 = f(X)^T \Delta f(X) \quad (43)$$

其中第一项是有监督的损失项 \Rightarrow 带了一个先验假设: 相邻顶点相似. 用 GCN 直接在有监督损失上训练可以避免这个先验.

4.3 Adaptive GCN/AGCN, [Li et al., 2018b]

Target 学习隐含的结点关系.

Idea 学习一个 residual Laplace 算子: $\hat{\mathbf{L}} = \mathbf{L} + \alpha \mathbf{L}_{res}$, \mathbf{L}_{res} 学习得到:

$$\mathbf{L}_{res} = \mathbf{I} - \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{\frac{1}{2}} \quad (44)$$

$$\hat{\mathbf{D}} = \text{deg}(\hat{\mathbf{A}}) \quad (45)$$

其中 $\hat{\mathbf{A}}$ 学得:

$$G_{x_i, x_j} = \exp(-D(x_i, x_j)/2\sigma^2) \quad (46)$$

$$D(x_i, x_j) = \sqrt{(x_i - x_j)^T M (x_i - x_j)} \text{ (Mahalanobis dist.)} \quad (47)$$

其中 M 是学得的参数, 满足 $M = \mathbf{W}_d^T \mathbf{W}_d$

4.4 Graph Neural Net(GNN)

Note ChebNet, GCN 可以看作是在图的 $r/1$ -hops 上作用, 是 GNN 的特例!

Idea 一个 GNN 在不同层上学一个高/低通算子: $f \rightarrow \mathbf{W}f, f \rightarrow \Delta f$.

对于每个顶点上一个 p -维向量的图信号 \mathbf{F} , GNN 考虑一个非线性函数 $\eta_\theta : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}^q$ 应用在每个图顶点上, $g_i = \eta_\theta((\mathbf{W}f)_i, (\mathbf{D}f)_i)$, 令 $\eta(a, b) = b - a$ 给出 Laplace 算子, η 的选择给出不同任务适用的 diffusion kernel.

- 连续应用同一个卷积层 $c_\theta = g_\theta(f) \Rightarrow$ 动力系统的稳态
- Coarsening \Rightarrow 单一输出
- 可用于实现其它图上的算子

5 Charting-based Methods

Problem 在不同域上进行同种操作? 但是定义的卷积难以在不同域上推广 \Rightarrow 定义另一种卷积!

考虑欧式卷积, 它涉及一个一个 pixel 移动 kernel 并作乘法, 图上对应? 结构没有平移不变性! 流形/图上缺少全局参数化 \Rightarrow 寻找内蕴坐标 \Rightarrow 图册和局部坐标系!

Idea 寻找权函数 $v_{1\dots J}(x, \cdot)$, 那么定义

$$D_j(x)f = \int_M f(x')v_j(x, x')dx' \quad (48)$$

这提供了内蕴的卷积

$$(f * g)(x) = \sum_j g_j D_j(x)f \quad (49)$$

这里 g 是应用在 patch 上的各系数, 参数个数 $O(J) = O(1)$.

5.1 Geodesic CNN(测地 CNN)

Idea 使用切空间上自带的极坐标

$$\rho(x') = d(x, x'), \theta(x') = \dots$$

那么权函数

$$v_{ij}(x, x') = e^{-(\rho(x') - \rho_i)^2 / 2\sigma_\rho^2} e^{-(\theta(x') - \theta_j)^2 / 2\sigma_\theta^2} \quad (50)$$

其中 $i = 1, \dots, J, j = 1, \dots, J'$ 是半径/角度的个数, 把 patch 拆分为了 JJ' 个 bin, 且大小为 $\sigma_\rho \times \sigma_\theta$

5.2 Anisotropic CNN(各向异性 CNN)

Idea 使用各向异性的扩散核:

$$f_t(x, t) = -\text{div}(A(x)\nabla f(x, t)) \quad (51)$$

其中 A 是热导率张量, 在 2d 情形时为 2×2 矩阵

一个选择:

$$A_{\alpha\theta} = R_\theta \begin{pmatrix} \alpha & \\ & 1 \end{pmatrix} R_\theta^T(x) \quad (52)$$

其中 R_θ 是按照某个轴 (如最大曲率轴) 的旋转. 对应的 heat kernel:

$$h_{\alpha\theta t}(x, x') = \sum_{i \geq 0} \exp(-t\lambda_{\alpha\theta i}) \phi_{\alpha\theta i}(x) \phi_{\alpha\theta i}(x') \quad (53)$$

其中 $\phi_{\alpha\theta 0}(x), \dots$ 是特征函数, $\lambda_{\alpha\theta i}$ 是对应特征值, of 各向异性 Laplace 算子 $\Delta_{\alpha\theta} f(x) = -\text{div}(A_{\alpha\theta}(x)\nabla f(x))$

5.3 Dynamic Graph CNN(DGCNN)

Idea 使用 MLP 学习一个权重函数 \Rightarrow EdgeConv block | 用 k-NN 建立点云的图表示.

对于两个点 $x_i, x_j \in \mathbb{R}^F$, 计算边权 $h_{\Theta}(x_i, x_j)$ (e.g. use MLP), 并在点上对相连的边做聚合操作 \square (some mean/sum/max etc.) 得到 F' 维的顶点特征:

$$x'_i = \square_{(i,j) \in \mathcal{E}} h_{\Theta}(x_i, x_j) \quad (54)$$

一个经典选择是 patch 上的卷积:

$$x'_{im} = \sum_m \theta_m \cdot x_j \quad (55)$$

m 是 channel 序号, 另几种:

$$h(x_i, x_j) = \begin{cases} h(x_i), \text{ or} \\ h(x_j), \text{ and Gaussian kernel (权重)} \\ h(x_j - x_i) \\ \bar{h}(x_i, x_j - x_i) \end{cases} \quad (56)$$

这种方法 (DGCNN) 里:

$$e'_{ijm} = \text{ReLU}(\theta_m \cdot (x_j - x_i) + \phi_m x_i) \quad (57)$$

$$x'_{im} = \max_j e'_{ijm} \quad (58)$$

Note 每次进行了一次卷积之后, 重新构建 (这里是 k-NN) 图结构 \Rightarrow 动态性. 并且, 这卷积操作具有形变不变性和平移不变性 (如果 $\phi \equiv 0$), 具体网络架构采用了 Dense-Conn. 思想 (之前的层全部连接到 fc 层上):

5.4 Mixture Model Network(MoNet, Monti et al.)

Idea 在图册上定义局部坐标系 $\mathbf{u}_x(x')$, 在这些坐标上应用一些带参数的 $\text{kernel}[v_{1 \dots J}(\mathbf{u})]$ 以形成权重函数.

采用 Gaussian 核:

$$v_j(\mathbf{u}) = \exp\left(-\frac{1}{2}(\mathbf{u} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{u} - \boldsymbol{\mu}_j)\right) \quad (59)$$

参数是 J 个 $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$ 和 $\boldsymbol{\mu} \in \mathbb{R}^{d \times 1}$.

实践中, 这些更大的自由度 \Rightarrow 很多问题的 SOTA 方法

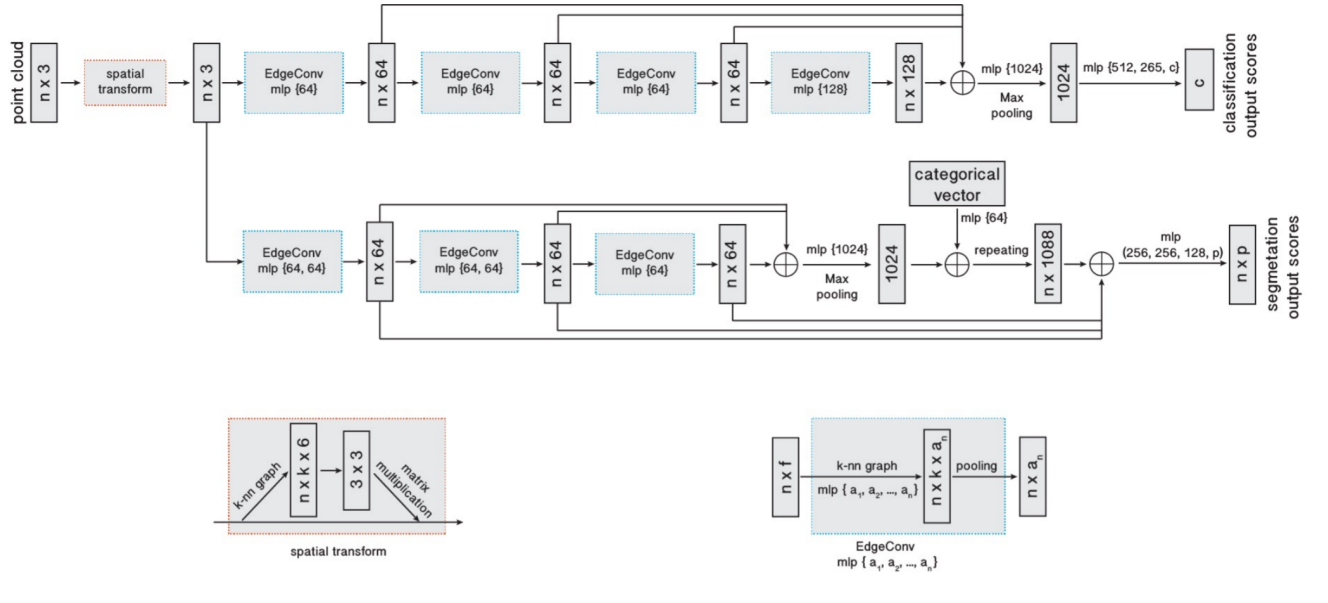


图 1: Structure of DGCNN

6 Spatial Methods

6.1 Neural FPS (Duvenaud et al., 2015)

Idea different node degrees \Rightarrow different weights

$$x = h_v^{t-1} + \sum_{i=1}^{|N_v|} h_i^{t-1} \quad (60)$$

$$h_v^t = \xi(x \mathbf{W}_t^{|N_v|}) \quad (61)$$

$$\text{where } \mathbf{W}_t^{|N_v|} \text{ is weights on layer } t \text{ of node deg. } |N_v| \quad (62)$$

Limitation Can't go large-scale

6.2 PATCHY-SAN (Niepert et al., 2016)

Idea 选择并正则化正好 k 个邻接点 (可以不是 1-hop 上的, 若不够), 这些点送到卷积层去进行计算

1. **Node Seq. Selection** 用某种图遍历过程将图排序, 然后选择一个处理图节点的顺序, 然后按照 stride s 选取结点, 直到选取了 w 个结点.(不一定包含所有结点)
2. **Neighborhood Assembly** 对于上述序列中的每个结点, 使用 BFS 得到 k 个邻接点.
3. **Graph Norm.** 试图给感受野中的结点一个顺序. 参见原论文!
4. **Conv. Architecture** 接着就可以送到 CNN 中, 其中正则化了的邻域就是感受野, 边/顶点特征维度就是 channels.

6.3 Diffusion-Convolutional NN/DCNN(Atwood and Towsley, 2016)

Idea 对于结点分类, 有

$$\mathbf{H} = \xi(\mathbf{W}^c \odot \mathbf{P}^* \mathbf{X}) \quad (63)$$

其中 $\mathbf{X} : N \times F$, $\mathbf{P}^* = \{\mathbf{P}^i\}_i : N \times K \times N$, \mathbf{P} 是度正则化的转移矩阵 (deg.-norm. transition mat.), \mathbf{H} 是图顶点的 diffusion-representation.

6.4 Dual GCN/DGCN(Zhuang and Ma, 2018)

Idea 分别考虑全局和局部一致性, 使用两个 CNN, 第一个即 GCN, 第二个将邻接矩阵替换成了 PPMI(positive pointwise mutual info.) 矩阵 (两个网络共享权重!):

$$\mathbf{H}' = \xi(\mathbf{D}_P^{-1/2} \mathbf{X}_P \mathbf{D}_P^{1/2} \mathbf{H} \Theta) \quad (64)$$

其中 \mathbf{X}_P 是 PPMI 矩阵, \mathbf{D}_P 是对应度矩阵.

Motivations of 2nd Net 类似 context 的结点应该有类似表示.

我们把这两个 CNN 分别称为 $Conv_A$ 和 $Conv_P$, 那么可以把两个网络集成在一个 loss 下

$$L = L_0(Conv_A) + \lambda(t) L_{reg}(Conv_A, Conv_P) \quad (65)$$

其中 $\lambda(t)$ 是动态平衡因子, $L_0(Conv_A)$ 是有监督的 loss(通过结点 label), 若网络的输出是 Z^A , softmax 化后是 \hat{Z}^A , 那么可以写出

$$L_0(Conv_A) = -\frac{1}{|y_L|} \sum_{l \in y_L} \sum_{i=1 \dots C} Y_{i,l} \ln(\hat{Z}_{i,l}^A) \quad (66)$$

另一边的 loss

$$L_{reg}(Conv_A, Conv_P) = \frac{1}{n} \sum_{i=1 \dots n} \|\hat{Z}_{i,:}^A - \hat{Z}_{i,:}^P\|^2 \quad (67)$$

6.5 Learnable GCN/LGCN(Gao et al., 2018)

Idea 基于 LGCL(Learnable Graph Conv. Layer) 和子图学习策略. LGCL 提升 CNN 作集成器, 在邻接矩阵上作 max-pooling, 得到 top-k 特征元素并送到 1-d CNN 上得到 repr., 具体 FP 过程:

$$\widehat{H}_t = g(H_t, A, k) \quad (68)$$

$$H_{t+1} = c(\widehat{H}_t) \quad (69)$$

其中 A 是邻接矩阵, g 是选择最大 k 个结点的 op, c 是卷积层.

关于 g : 给定结点 x , 设它有 n 个邻接点, 特征是 c 维的,

7 Spatial-Spectral Methods

7.1 Windowed Fourier Transform

Idea 传统傅里叶变换由不确定性原理限制, 时域局部性 \Rightarrow 难以达成频域局域性 \Rightarrow 带窗傅里叶变换 (Windowed Fourier Transform, WFT, Short-Time FT, Spectrogram)

WFT: (on \mathbb{R}^n , $g(x)$ 是窗口函数)

$$(Sf)(x, \omega) = \int_{\mathbb{R}} f(x') g(x' - x) e^{-i\omega x'} dx' \quad (70)$$

$$= \langle f, g_{x, \omega} \rangle_{L^2(\mathbb{R})} \quad (71)$$

$$\text{where } g(x' - x) e^{-i\omega x'} = g_{x, \omega}(x') \quad (72)$$

拓展到非欧域上, 要定义平移算子为和 δ 函数的卷积:

$$(g * \delta_{x'})(x) = \sum_{i \geq 0} \langle g, \phi_i \rangle_{L^2(M)} \langle \delta_{x'}, \phi_i \rangle_{L^2(M)} \phi_i(x) \quad (73)$$

$$= \sum_{i \geq 0} \hat{g}_i \phi_i(x') \phi_i(x) \quad (74)$$

那么就可以写出非欧域上的 WFT atoms:

$$g_{x', j}(x) = \phi_j(x') \sum_{i \geq 0} \hat{g}_i \phi_i(x') \phi_i(x) \quad (75)$$

这里 window 大小被 Fourier 系数 \hat{g}_i 确定. 给出 WFT:

$$(Sf)(x', j) = \langle f, g_{x', j} \rangle_{L^2(M)} = \sum_{i \geq 0} \hat{g}_i \phi_i(x') \langle f, \phi_i \phi_j \rangle_{L^2(M)} \quad (76)$$

Note 这些定义都是内蕴的!

7.2 Wavelet Transform

Idea 频域分解 \Rightarrow 尺度分解 \Rightarrow 小波分解

Target 定义时/频域上局域性都很好的小波基, 可用于还原/近似信号
一个简单实现: Haar 小波.

Learning: 找到个尺度上的最优的结点配对 \Rightarrow 多项式复杂度!

7.3 Localized Spectral CNN (LSCNN, Boscaini et al.)

Idea 用 (48), (49) 之中的卷积, 但是利用 WFT 定义 patch:

$$D_j(x)f = (Sf)(x, j) \quad (77)$$

8 Application

- 网络分析
 - 文献引用图 (CORA)
 - ranking & community detection | 求解特征值/Fiedler 向量, 包含了最小割的信息.
 - PageRank: 通过借一个改动的 Laplace 算子的特征值来 rank.
- 推荐系统
 - 矩阵补全问题 \Rightarrow Geometric CNN
 - (Monti et al.) Multi-Graph CNN + RNN: 由 MGCNN 提取的特征送到 RNN 作为对 score 的更新.
- CV/CG
 - 3d-geometric data processing: rasterizing 可行, 但是会丢失拓扑信息
 - CG(Mesh Correspondence/Classification)
- 粒子物理和化学
- 分子设计
- 医学影像

9 Open Problems and Discussions

- Generalization
 - 谱方法: 难以跨域
 - 时域方法/图册方法: 定义适当的局部坐标系较难
- 时变域: 动态形状, abnormal social events
- 有向图: Laplace 算子不是对称的, 没有正交特征值分解
- 生成问题: 如何从内蕴的结构/表示重建几何结构
- 计算效率问题: 非欧域上的 DL 常常没有网格结构, 难以在 GPU 上高效计算