SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS

Томинин Ярослав Дмитриевич

МФТИ

15 апреля 2019 г.

Постановка задачи

- Граф $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
  - $|\mathcal{V}| = N$
  - Каждая вершина v_i- социальный объект(пример:человек, статья)
  - Ребра $(v_i, v_j) \in \mathcal{E}$ - связи между объектами
- $X \in \mathbb{R}^{N \times F}$, где$|X_i| = F -$ кол-во признаков v_i
- $A \in \mathbb{R}^{N \times N}$ - матрица смежности
- $D \in \mathbb{R}^{N \times N}$, где $D_{ii} = \sum\limits_j A_{ij}$

Предположение

v_i и v_j с общими соседями обладают сходными свойствами

Требуется

Решить задачу классификации на полуразмеченных данных

Мотивация

Структурированные данные

Есть метрика или геометрические свойства?

да

нет

Картинки, видео

CNN-очень мощный, универсальный подход

Данные из социальных сетей, статьи,интернет

Подходы

Пространственная конструкция

Не универсальна

Спектральная подход

Есть потенциал

Идея обобщения

Предпосылки

Хочется обощить модели нейронных сетей на случай структурированных данных, не обладающих геометрическими свойствами(метрикой).

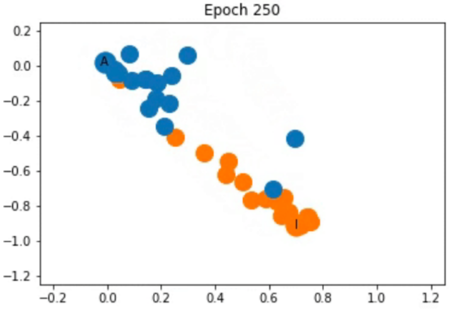
Предлагаемый подход-архитектура GCN

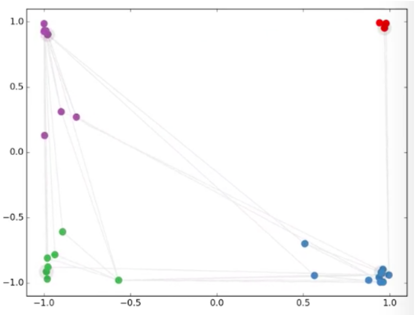
- Каждый слой GCN имеет вид $H^{(l+1)} = f(H^{(l)}, A)$, где
  - $H^{(0)} = X$
  - $H^{(L)} = Z$
  - L-количество слоев
  - X-матрица признаков
  - в статье $f(H^{(l)}, A) = \sigma(\tilde{D}^{-0.5}\tilde{A}\tilde{D}^{-0.5}H^{(l)}W^{(l)})$, σ-функция активации(ReLU)

Презентация посвящена обоснованию этого подхода

Пример задачи

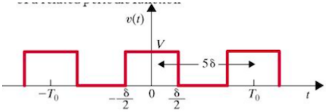
Zachary’s клуб карате
Произошел конфлик и появилось разделение на общество инструктора и общество администратора.
Инструктор и администратор помечены буквами



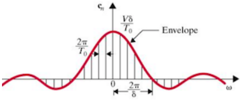


Спектральное разложение

- $f(x) \in C[-\infty, +\infty]$, T_0-период

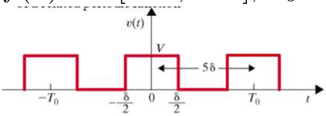


$\Rightarrow$

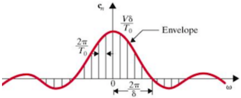


Спектральное разложение

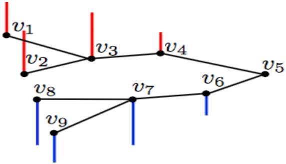
- $f(x) \in C[-\infty, +\infty]$, T_0-период



$\Rightarrow$



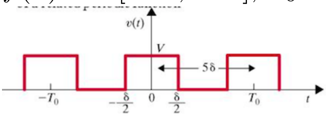
- Граф $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $f : V \to \mathbb{R}^N$



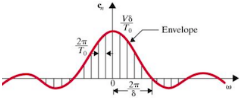
$\Rightarrow$

Спектральное разложение

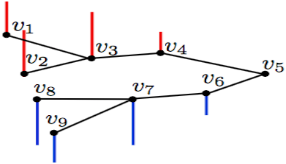
- $f(x) \in C[-\infty, +\infty]$, T_0-период



$\Rightarrow$



- Граф $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $f : V \to \mathbb{R}^N$



$\Rightarrow$

Требуется

Ввести понятие частоты для графа

Лапласиан графа

- A - матрица смежности,$D : D_{ii} = \sum_j A_{ij}$

Лапласиан графа

- A - матрица смежности,$D : D_{ii} = \sum_j A_{ij}$
- $L = I_N - D^{-0.5} A D^{-0.5}$
    - симметричен
    - раскладывается в базис СВ $U : L = U \Lambda U^T$
    - $\forall i, \lambda_i \geq 0$

Лапласиан графа

- A - матрица смежности,$D : D_{ii} = \sum_j A_{ij}$
- $L = I_N - D^{-0.5} A D^{-0.5}$
  - симметричен
  - раскладывается в базис СВ $U : L = U \Lambda U^T$
  - $\forall i, \lambda_i \geq 0$
- Определим $f : V \to \mathbb{R}^N$

Лапласиан графа

- A - матрица смежности,$D : D_{ii} = \sum_j A_{ij}$
- $L = I_N - D^{-0.5} A D^{-0.5}$
  - симметричен
  - раскладывается в базис СВ $U : L = U \Lambda U^T$
  - $\forall i, \lambda_i \geq 0$
- Определим $f : V \to \mathbb{R}^N$
- $Lf = \sum_{i,j} A_{ij}(f(v_i) - f(v_j))$

Очень похож на дифференциальный оператор

Преобразование Фурье для графа

Мотивация

Очень часто мы ничего не знаем о функции и хотим получить какие-то свойства. Поэтому полезно перейти в область Фурье, выделить фильтром нужные свойства и вернуться обратно.

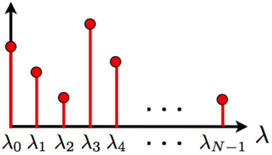
- $\mathcal{G} = (\mathcal{V}, \mathcal{E}), f : \mathcal{V} \to \mathbb{R}^N$, $U : L = U \Lambda U^T$

Определение

Преобразование Фурье сигнала отображает $f(\mathcal{V}) \to U^T f(\mathcal{V})$

- i-ая кооодината полученного вектора

$$[U^T f(\mathcal{V})]_i = [\hat{f}(\mathcal{V})]_i = \langle U_i, f(\mathcal{V}) \rangle$$



Почему выбрали Лапласиан?

$f : \mathbb{R} \to \mathbb{R}$

- Оператор Лапласа: $\dfrac{d^2}{dx^2}$
- Собственная функция :$e^{i\omega x}$
- Обычное преобразование Фурье:
- $$\hat{f}(\omega) = \int e^{i\omega x^*} f(x)dx$$
- $$f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{i\omega x} d\omega$$

$f : \mathcal{V} \to \mathbb{R}$

- Лапласиан графа: L
- Собственный вектор :U_i
- Преобразование Фурье для графа:
- $$\hat{f}(i) = \langle U_i, f(\mathcal{V}) \rangle = \sum_{j=1}^{N} U_{ij}^* f(v_j)$$
- $$f(j) = \sum_{i=0}^{N-1} U_{ij}^* \hat{f}(v_i)$$

Почему выбрали Лапласиан?

$f : \mathbb{R} \to \mathbb{R}$

- Оператор Лапласа: $\dfrac{d^2}{dx^2}$
- Собственная функция :$e^{i\omega x}$
- Обычное преобразование Фурье:
- $$\hat{f}(\omega) = \int e^{i\omega x^*} f(x)dx$$
- $$f(x) = \frac{1}{2\pi} \int \hat{f}(\omega) e^{i\omega x} d\omega$$

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

$f : \mathcal{V} \to \mathbb{R}$

- Лапласиан графа: L
- Собственный вектор :U_i
- Преобразование Фурье для графа:
- $$\hat{f}(i) = \langle U_i, f(\mathcal{V}) \rangle = \sum_{j=1}^{N} U_{ij}^* f(v_j)$$
- $$f(j) = \sum_{i=0}^{N-1} U_{ij}^* \hat{f}(v_i)$$

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

Фильтр $g_\Theta(\Lambda)$, $y = g_\Theta \star x$

Проблемы

1)Параметр на является K-локализованным
2)Сложность обучения $O(n)$

Решение 1-ой проблемы

$$g_\Theta(\Lambda) = \sum_{k=0}^{K-1} \Theta_k \Lambda^k, \text{ где } \Theta \in \mathbb{R}^K$$

- Рассмотрим отклик j-ой компоненты вектора на δ_i

$$(U^T(g_\Theta(\Lambda))U\delta_i)_j = ((g_\Theta(L))\delta_i)_j = \sum_k \Theta_k (L^k)_{ij} \tag{1}$$

Фильтр $g_\Theta(\Lambda)$, 1-ая проблема

Лемма о локализованности

Пусть $v_i, v_j \in \mathcal{V}$ и $d(v_i, v_j) > K$ тогда $\Rightarrow (L^K)_{ij} = 0$

Получается, что (1) учитывает только вершины $v_j : d(v_i, v_j) < K$. Следовательно, фильтр является локализованным.

Умеем

- Получать фильтр за $O(K)$
- Фильтр $g_\Theta(\Lambda)$ K-локализован

Сложности

- Фильтрация сигнала $x \to y : y = U g_\Theta(\Lambda) U^T x$ стоит $O(n^2)$(умножение x на U)

Фильтр $g_\Theta(\Lambda)$, 2-ая проблема

План действий

- Представим функцию $g_\Theta(\Lambda)$ в виде $g_\Theta(\Lambda) = \sum_{k=0}^{K} \Theta_k T_k(\tilde{\Lambda})$
- Для $T_k(\tilde{\Lambda})$ будет дана рекурсивная формула
- Можем вычислить $T_k(\tilde{L})x$ за k умножений $\tilde{L}$ на x. То есть за $O(k|\mathcal{E}|)$

В перспективе

Если получится найти такую рекурсивную формулу, то вместо умножения Ux фильтрация будет иметь вид:$U^T(g_\Theta(\Lambda))Ux = U^T(\sum_{k=0}^{K} \Theta_k T_k(\tilde{\Lambda}))Ux = (\sum_{k=0}^{K} \Theta_k T_k(\tilde{L}))x$

Стоимость $T_k(\tilde{L})$ равна $O(k|\mathcal{E}|) \Rightarrow$ вся фильтрация стоит $O(k|\mathcal{E}|)$

Фильтр $g_\Theta(\Lambda)$, 2-ая проблема

- Чтобы приблизить нашу функцию многочленами Чербышева, нужно линейно отобразить область определения в $[-1, 1]$

$$\tilde{\Lambda} = \frac{2\Lambda}{\lambda_{max}} - I_N$$

- Представим функцию $g_\Theta(\Lambda)$ в виде $g_\Theta(\Lambda) = \sum_{k=0}^{K} \Theta_k T_k(\tilde{\Lambda})$, где Θ'_k- вектор коэффициентов Чебышева

Рекуррента

$$\boxed{\tilde{x}_k = 2\tilde{L}\tilde{x}_{k-1} - \tilde{x}_{k-2}}$$, где $\tilde{x}_k = T_k(\tilde{L})x$, $\tilde{x}_0 = x, \tilde{x}_1 = \tilde{L}x$

Соотношение существующих подходов

Полуконтролируемое обучение на графах

Пространственная конструкция

Спектральный подход

Не универсальна

Приближение с помощью полиномов Чебыщева

K=1

K»1
Достаточно мощная

Очень много обучаемых параметров

Ограничимся линейным фильтром $g_\Theta(\Lambda)$

Рассмотрим $K = 1$

Предположим, что $\lambda_{max} = 2$

Тогда наш фильтр приймет вид: $g_{\Theta'} \star x \approx \Theta'_0 x - \Theta'_1 D^{-0.5} A D^{-0.5} x$

Недостатки

- Модель сильно ограничена из за линейного приближения и плохо работает на регулярных графах(решетка)

Преимущества

- Быстрые вычисления
- Мало обучаемых параметров, можно строить больше уровней

Вывод

Подход не является универсальным в силу его ограниченности, но при этом позволяет решать большой спектр задач.

Ограничение на обучаемые параметры фильтра

Соображение

Часто пытаются уменьшить количество параметров, чтобы избежать переобучения и сделать операции менее затратными.

Рассмотрим $\Theta'_0 = -\Theta'_1$

В итоге получим формулу $g_\Theta \star x \approx \Theta x + \Theta D^{-0.5} A D^{-0.5} x$

Соображение

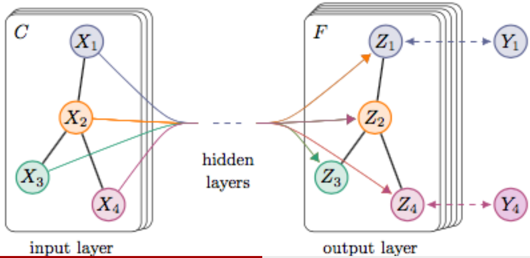
Сейчас $\lambda_i \in [0, 2]$. Это может мешать обучению при обратном распространении ошибки, поэтому нормализуем наш фильтр.

$I_N + D^{-0.5} A D^{-0.5} \to \tilde{D}^{-0.5} \tilde{A} \tilde{D}^{-0.5}$, где $\tilde{A} = A + I_N$, $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$

В результате $g_\Theta \star x \approx \tilde{D}^{-0.5} \tilde{A} \tilde{D}^{-0.5} \Theta x$

Реализация GCN

- Построим GCN с двумя слоями
  - $f(H^{(l)}, A) = ReLU(\tilde{D}^{-0.5}\tilde{A}\tilde{D}^{-0.5}H^{(l)}W^{(l)})$
  - фукция кросс-энтропийной ошибки $\mathcal{L} = -\sum_{l}\sum_{f=1}^{F} Y_{lf} \ln Z_{lf}$, где $F = |Z_i|$,Y_{lf}-метка вершины
  - После двух слоев идет функция softmax $softmax(x_i) = \frac{\exp(x_i)}{\sum_i \exp(x_i)}$



Обучение

- Веса нейронной сети $W^{(0)}, W^{(1)}$ обучаются с помощью градиентного спуска.