Semi-Supervised Classification with Graph Convolutional Networks

Thomas N. Kipf, Max Welling

ВШЭ ФКН ПМИ
Юрлов Павел

21 ноября 2019

План

1. Задача
2. Архитектура модели
   - Свёртки на графах
3. Semi-supervised learning
4. Эксперименты
   - Классификация вершин
   - Представления вершин
   - Влияние глубины сети на качество
5. Ограничения
6. Выводы

Задача

Semi-supervised на графах

- Задача классификации вершин графа, где доступны метки лишь малого подмножества вершин
- Классификация с частичным привлечением учителя
- Примеры: граф цитирования, социальные сети

Semi-supervised на графах: возможный подход

- Сеть принимает на вход только представления вершин, без структуры графа.
  Модификация функции потерь для учёта неразмеченных вершин, например, с использованием лапласиана графа:

$$\mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_{reg},$$

  где

$$\mathcal{L}_{reg} = \sum_{i,j} A_{i,j} ||f(X_i) - f(X_j)||^2 = f(X)^T \Delta f(X), \ \Delta = D - A$$

- Авторы статьи поступают иначе, кодируя структуру графа сетью $f(X, A)$

Архитектура модели

Обозначения

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

- Вход: $X \in \mathbb{R}^{N \times D}$ (N вершин, D входных признаков)
  $A \in \mathbb{S}_+^N$ — матрица смежности
- Выход: $Z \in \mathbb{R}^{N \times F}$ (F выходных признаков)
- Слой сети: $H^{(l+1)} = f(H^{(l)}, A)$
  $H^{(0)} = X,\ H^{(L)} = Z$ (L — число слоёв)

Слой сети: простое объяснение

- Прямой проход по слою: $H^{(l+1)} = \sigma(A\ H^{(l)}\ W^{(l)})$
- Добавим петли: $\widetilde{A} = A + I_N$
- Нормализуем матрицу смежности:
  $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$
  $\widetilde{A} = \widetilde{D}^{-1}\ \widetilde{A}$
  Но лучше так: $\widetilde{A} = \widetilde{D}^{-1/2}\ \widetilde{A}\ \widetilde{D}^{-1/2}$
- Получаем:

$$H^{(l+1)} = \sigma(\widetilde{D}^{-1/2}\ \widetilde{A}\ \widetilde{D}^{-1/2}\ H^{(l)}\ W^{(l)}) \qquad (*)$$

Спектральная свёртка на графе

$$g_\theta \star x = U g_\theta U^T x \tag{1}$$

- $x \in \mathbb{R}^N$ — одноканальный сигнал
- $g_\theta = diag(\theta)$ с параметром $\theta \in \mathbb{R}^N$ из преобразования Фурье
- U — матрица собств. векторов нормализованного лапласиана графа $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^T$
- Проблема: умножение на U и собственное разложение вычислительно сложны

Приближение многочленами Чебышёва

$$\left[g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta'_k T_k(\widetilde{\Lambda}) \right]$$

$$g_{\theta'} \star x \approx \sum_{k=0}^{K} \theta'_k T_k(\widetilde{L}) x \tag{2}$$

- Масштабирование: $\widetilde{\Lambda} = \frac{2}{\lambda_{max}}\Lambda - I_N,\ \widetilde{L} = \frac{2}{\lambda_{max}}L - I_N$
- Многочлены Чебышёва:
  $T_0(x) = 1,\ T_1(x) = x$
  $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$
  (Легко запомнить: $T_k(\cos\alpha) = \cos(k\alpha)$)
- K-локализация

Ограничение многочленов до первого порядка

$$K = 1: \quad g_{\theta'} \star x \approx \theta'_0 x + \theta'_1 (\frac{2}{\lambda_{max}} L - I_N) x \tag{3}$$

- Мы получили линейную функцию от L
- Из K-локализации получили зависимость лишь от непосредственных соседей, но это ограничение преодолевается добавлением слоёв
- Кроме того, мы не обязаны использовать именно многочлены Чебышёва (см. слайд 21)

Дальнейшие приближения

- Изменение масштаба; параметры одинаковы для всего сигнала x

$$\lambda_{max} \approx 2: \quad g_{\theta'} \star x \approx \theta'_0 x + \theta'_1 (L - I_N)x = \theta'_0 x + \theta'_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x \quad (4)$$

- Ещё уменьшим число параметров

$$\theta'_0 = \theta, \theta'_1 = -\theta: \quad g_\theta \star x \approx \theta(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}})x \quad (5)$$

- Ренормализация $\widetilde{A} = A + I_N$, $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$:

$$g_\theta \star x \approx \theta \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} x \quad (6)$$

Итоговый слой

Обобщим:

$$Z = \sigma(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}X\Theta), \tag{7}$$

где $X \in \mathbb{R}^{N\times C}$ — вход слоя с C каналами,
$\Theta \in \mathbb{R}^{C\times F}$ — параметры фильтров,
$Z \in \mathbb{R}^{N\times F}$ — выход слоя с F каналами,
$\sigma(\cdot)$ — нелинейность.

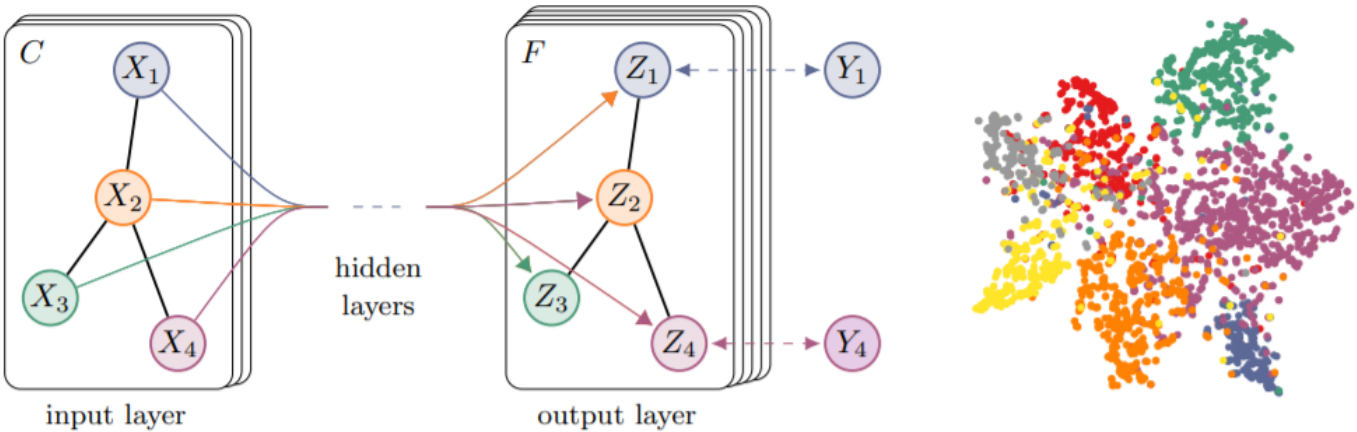
Вычислительная сложность $\mathcal{O}(|\mathcal{E}|CF)$ из-за эффективной реализации умножения на разреженную матрицу $\widetilde{A}$

Semi-supervised learning

Semi-supervised learning

1. Модель использует и матрицу представлений вершин X, и матрицу смежности A, которые перемножаются при прямом распространении, причём параметры фильтров одинаковы для всех вершин
2. Поэтому вершины с общими или похожими соседями (которые скорее всего из одного класса) получают близкие представления в дальнейших слоях
3. Функция потерь — кросс-энтропия только на вершинах с известными метками

Визуализация



(a) Graph Convolutional Network

(b) Hidden layer activations

Figure 1: *Left*: Schematic depiction of multi-layer Graph Convolutional Network (GCN) for semi-supervised learning with C input channels and F feature maps in the output layer. The graph structure (edges shown as black lines) is shared over layers, labels are denoted by Y_i. *Right*: t-SNE (Maaten & Hinton, 2008) visualization of hidden layer activations of a two-layer GCN trained on the Cora dataset (Sen et al., 2008) using 5% of labels. Colors denote document class.

Эксперименты

Архитектура сети и её обучение

1. 2-слойная сеть
   Предварительно вычисляем $\widehat{A} = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$
   Прямой проход ($W^{(0)} \in \mathbb{R}^{C\times H}$, $W^{(0)} \in \mathbb{R}^{H\times F}$):

$$Z = f(X, A) = \mathrm{softmax}\big(\widehat{A}\ \mathrm{ReLU}\big(\widehat{A}\ X\ W^{(0)}\big)W^{(1)}\big)$$

2. Функция потерь: $\mathcal{L} = -\sum_{l\in\mathcal{Y}_L}\sum_{f=1}^{F} Y_{lf}\ln Z_{lf}$
3. Градиентный спуск, на каждой итерации батч составляет весь датасет
4. Стохастичность при обучении из-за использования dropout
5. Память $\mathcal{O}(|\mathcal{E}|)$ благодаря разреженному хранению матрицы смежности

Датасеты

- Citation networks: 20 примеров с метками на каждый класс
- NELL: 1 пример с меткой на каждый класс

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001

Сравнение с бейзлайнами

Table 2: Summary of results in terms of classification accuracy (in percent).

Method	**Citeseer**	**Cora**	**Pubmed**	**NELL**
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

Сравнение разных методов

Description		Propagation model	Citeseer	Cora	Pubmed
Chebyshev filter (Eq. 5)	$K = 3$	$\sum_{k=0}^{K} T_k(\tilde{L})X\Theta_k$	69.8	79.5	74.4
	$K = 2$		69.6	81.2	73.8
1^{st}-order model (Eq. 6)		$X\Theta_0 + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}X\Theta_1$	68.3	80.0	77.5
Single parameter (Eq. 7)		$(I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})X\Theta$	69.3	79.2	77.4
Renormalization trick (Eq. 8)		$\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta$	**70.3**	**81.5**	**79.0**
1^{st}-order term only		$D^{-\frac{1}{2}}AD^{-\frac{1}{2}}X\Theta$	68.7	80.5	77.8
Multi-layer perceptron		$X\Theta$	46.5	55.1	71.4

Время обучения (с учителем, на случайном графе)

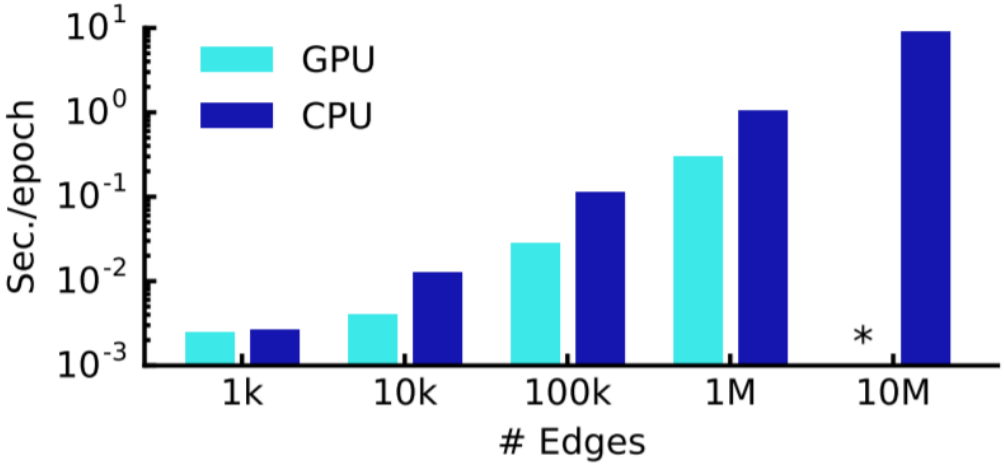
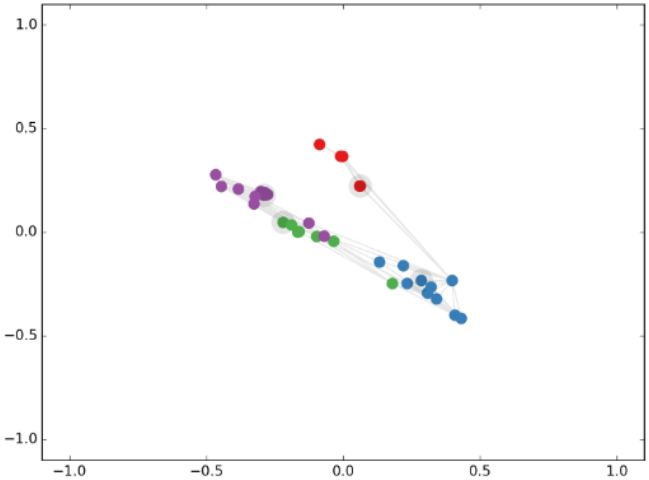


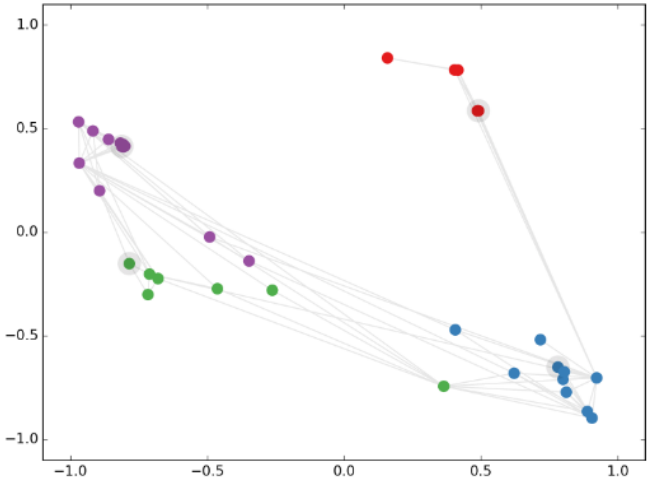
Figure 2: Wall-clock time per epoch for random graphs. (*) indicates out-of-memory error.

Представления вершин 1

- Граф (karate club network): $|\mathcal{V}| = 34, |\mathcal{E}| = 154$, модулярная кластеризация на 4 класса
- 3-слойная сеть с tanh активацией, 2 выходных канала, 1 размеченный пример на класс

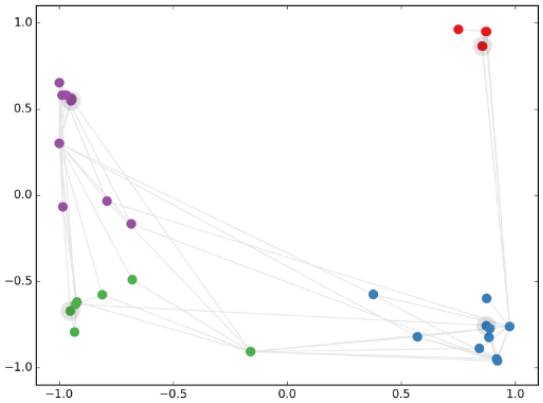


(a) Iteration 25

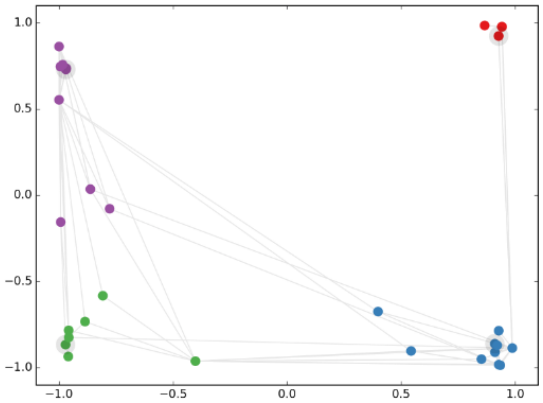


(b) Iteration 50

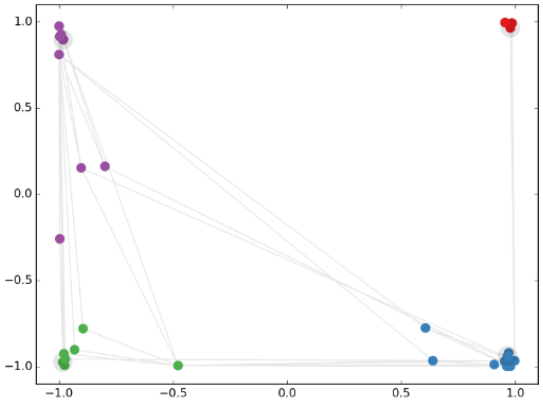
Представления вершин 2



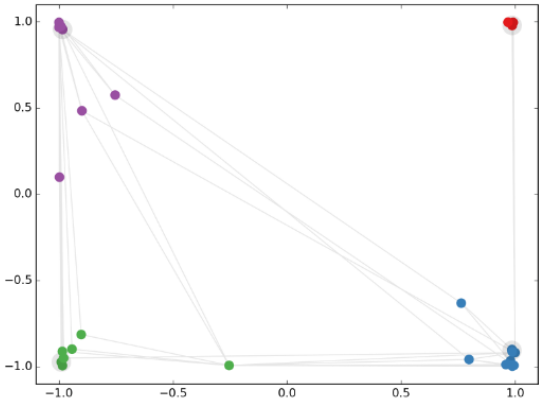
(c) Iteration 75



(d) Iteration 100



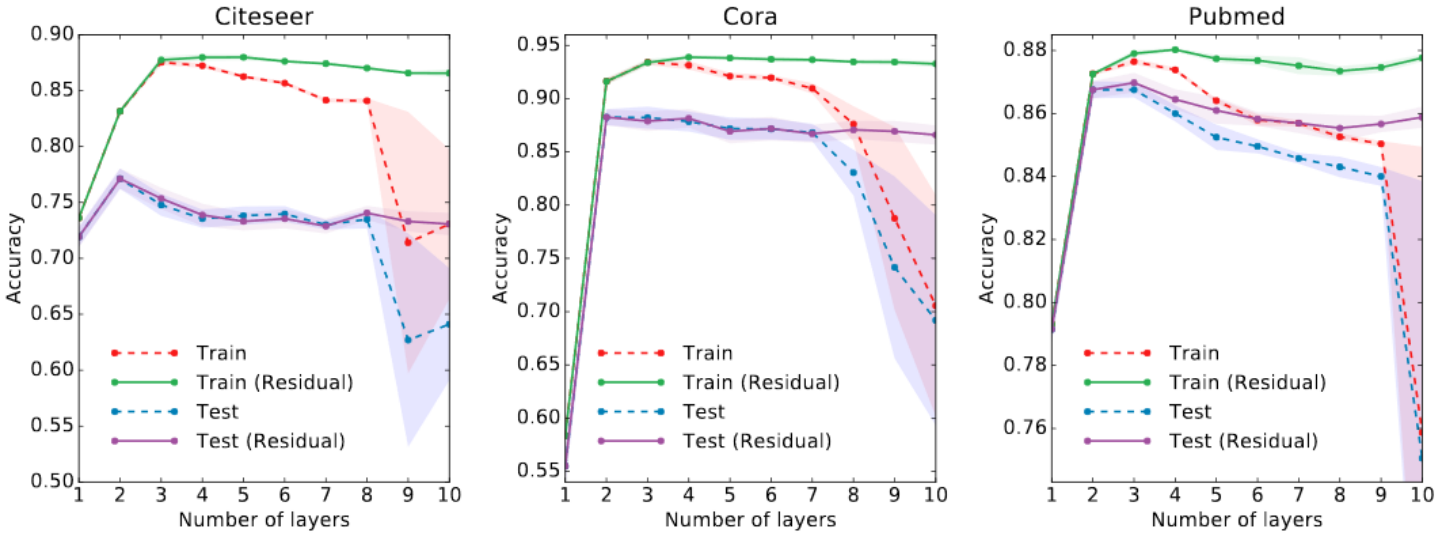
(e) Iteration 200



(f) Iteration 300

Влияние глубины при обучении с учителем

$$H^{(l+1)} = \sigma(\widetilde{D}^{-\frac{1}{2}}\ \widetilde{A}\ \widetilde{D}^{-\frac{1}{2}}\ H^{(l)}\ W^{(l)})\ (+H^{(l)}, \text{if residual})$$



Ограничения

Ограничения модели и их возможное преодоление

1. Из-за обучения на всём датасете на каждой итерации требования по памяти растут линейно от размера датасета, точнее числа рёбер. Но использование мини-батчей должно учитывать зависимость от соседей k-го порядка, в т. ч. не из мини-батча, на k-ом слое
2. Направленные рёбра и признаки рёбер не учитываются
3. Из-за произведённых приближений подразумевается локальность (зависимость от соседей до k-го порядка в k-слойной сети) и одинаковая важность петель и остальных рёбер. Последнее можно исправить, введя параметр λ: $\widetilde{A} = A + \lambda I_N$

Выводы

Выводы

- Вычислительно эффективный конволюционный слой, полученный приближением первого порядка спектральной свёртки на графе
- Модель способна учитывать как характеристики вершин, так и структуру графа
- Это позволяет ей довольно успешно справляться с классификацией вершин графа с частичным привлечением учителя

Вопросы

1. Запишите формулу $(*)$ свёрточного слоя модели, поясните обозначения.
2. Учитывает ли модель явным образом примеры без меток в функции потерь? Как модель учитывает их при обучении?
3. Какова сложность модели по памяти при обучении? Почему её нельзя понизить без модификации модели?

Источники

- https://arxiv.org/abs/1609.02907
- https://tkipf.github.io/graph-convolutional-networks/