

Литературный обзор по теме «Разработка  
алгоритма обучения с переносом для графовых  
структур данных»

Андреева Полина

15 июня 2020 г.

## Введение

Графовое представление данных это естественный, понятный человеку с одной стороны, и удобный, подходящий для машинной обработки с другой, способ визуализации данных из различных областей: от генетики до банковского дела. И хотя большинство самых распространенных архитектур нейронных сетей (например CNN, RNN) ориентировано на векторные структуры, в последнее время появились новые подходы глубокого обучения для представления и моделирования графово-структурированных данных.

Большинство успехов в глубоком обучении достигнуто с применением одной из двух парадигм - обучения с учителем (контролируемое обучение) или обучение с подкреплением. В обоих случаях пределы обучения определяются людьми. Для истинного интеллекта необходимы более независимые стратегии обучения. Например, младенцы изучают мир из любопытства, через наблюдение. Существует одна парадигма, которая углубляет понимание интеллекта - обучение без учителя, когда агент обучается с целью обучиться. Во взрослой жизни, люди постоянно приспосабливаются к текущей ситуации, используя имеющиеся навыки и знания. Такая адаптация позволяет не обучаться с нуля какому-либо ремеслу, если уже имеется опыт в похожих сферах, а лишь разобраться в относительно небольшом объеме специфических деталей. В связи с чем в глубоком обучении придумана следующая парадигма, углубляющая понимание работы истинного интеллекта и развивающая работу искусственного - обучение с переносом. С точки зрения затраченных ресурсов, такой способ решения задач намного эффективней, так как не требует большого объема новых данных.

Формальное определение обучения с переносом, сформулированное в [1], будет дано в следующем пункте. А в последующих разделах будут рассмотрены алгоритмы обучения с переносом на графовых структурах данных.

# Обучение с переносом

## Постановка задачи

Строгое определение задачи обучения с переносом, сформулированное S. Pan и Q. Yang [1]:

Домен  $\mathcal{D}$  состоит из пространства признаков  $\mathcal{X}$  и маргинального вероятностного распределения  $P(X)$ , где  $X = \{x_1, \dots, x_n\} \in \mathcal{X}$ . При заданном домене  $\mathcal{D} = \{\mathcal{X}, P(X)\}$ , задача  $\mathcal{T}$  задается также двумя компонентами: пространство меток  $\mathcal{Y}$  и целевая предсказательная функция  $f(\cdot)$ , которая находится из тренировочных данных, состоящих из пар  $\{x_i, y_i\}$ , где  $x_i \in X$  и  $y_i \in \mathcal{Y}$ .

В обучении с переносом существует два домена  $\mathcal{D}_s, \mathcal{D}_t$  и две задачи  $\mathcal{T}_s, \mathcal{T}_t$  - исходный и целевой. Задача: улучшить обучение целевой предсказательной функции  $f_T(\cdot)$  в целевом домене, используя знания в исходных  $\mathcal{D}_s, \mathcal{T}_s$ .

## Классификация по подходу переноса

В зависимости от того одинаковые ли или различаются признаки/распределения вероятностей/метки существуют несколько подходов в обучении с переносом:

### 1. Inductive transfer learning

Исходный и целевой домен один и тот же, а исходная и целевая задача различаются. Идея состоит в применении того набора допущений, который был использован для решения в исходном домене, при решении целевой задачи. В зависимости от того содержит ли исходный домен помеченные данные или нет, данный тип обучения с переносом делится на две соответствующие подкатегории.

### 2. Unsupervised transfer learning

Исходный и целевой домен один и тот же, а исходная и целевая задача различаются. Фокус направлен на неконтролируемое обучение в целевой задаче. Соответственно данные не помечены ни в одном из доменов.

### 3. Transductive transfer learning

Исходная и целевая задача похожи, но соответствующие домены различаются. Исходный домен содержит помеченные данные, целевой - нет. В зависимости от того, отличается ли пространство

признаков или распределения вероятностней у исходного и целевого домена, можно классифицировать задачи далее.

## **Классификация по предмету переноса**

Существует несколько важных вопросов, на которое необходимо ответить прежде чем приступить к разработке алгоритма для задач обучения с переносом: что переносить? как переносить? Рассмотренные в предыдущем пункте подходы переноса отвечают на последний вопрос. А в данном разделе будет рассмотрена классификация задач, в зависимости от ответа на первый вопрос - что переносить?

### **1. Instance transfer**

Определенная часть данных в исходном домене может быть передана для обучения в целевой домен посредством переназначения весов.

### **2. Parameter transfer**

Идея основана на предположении, что модели имеют некоторые общие параметры или априорные распределения гиперпараметров, которые и переносятся из исходной задачи в целевую.

### **3. Feature representations transfer**

Идея состоит в том, чтобы минимизировать расхождения в исходном и целевом домене посредством определения «хорошего» представления признаков (например «хороших» эмбдингов), которое и передается из исходного и в целевой домен.

### **4. Relational Knowledge transfer.**

Используется для данных, которые не независимы и не одинаково распределены, т.е. для таких, у которых для каждой точки есть связь с другой точкой. Предполагается, что некоторые отношения между данными аналогичны в целевом и исходном домене. Эти соотношения и есть знания, которые передаются.

В таблице 1 показаны какие случаи обучения с переносом могут быть применены в каждом подходе.

	Inductive Transfer Learning	Transductive Transfer Learning	Unsupervised Transfer Learning
<i>Instance-transfer</i>	✓	✓	
<i>Feature-representation-transfer</i>	✓	✓	✓
<i>Parameter-transfer</i>	✓		
<i>Relational-knowledge-transfer</i>	✓		

Таблица 1 [1]

## Обучение с переносом на графах

Графы – структуры, состоящие из уникальных сущностей (узлов графа) и связей между ними (ребер графа), являются основной формой представления и хранения знаний и удобным инструментом для работы с большинством видов данных. Среди основных преимуществ графового представления данных это логическая строгость и масштабируемость на большие объемы информации. Последнее время было сделано много попыток расширить сверточные нейронные сети на графовые структуры данных. Проблема графовых структур данных в самом не-векторном представлении. В связи с чем, основным подходом в задачах на графах стало представление элементов графа в некотором высокоразмерном пространстве, в виде векторов, так называемых эмбедингов [2]. В таком виде задача решается как для любой другой обычной сверточной нейронной сети. Методы, использующие эмбединги, называются Graph Neural Networks (GNNs) [2] и были применены к различных задачам относящимся к графикам.

Задачи на графах можно разделить на следующие наиболее общие группы:

- Предсказание связи между вершинами
- Классификация вершин
- Классификация ребер
- Классификация графов

## Различные методы обучения с переносом на графах

### Обучение с переносом внутренней геометрии на графово - структурированных данных [9]

*Подход:* Inductive. Feature representation transfer.

*Какую проблему решает:* Методы CNN и RNN ориентированы на матричные структуры данных. А данный подход улучшает глубокое обучение данных с графовой структурой с помощью обучения с переносом. Передавая внутреннюю геометрическую информацию, полученную в исходной области, этот метод позволяет построить модель для новой, но связанный задачи в целевой области без сбора новых данных и без обучения новой модели с нуля.

*Основная идея:* Обучить сверточную нейронную сеть на исходных данных, а затем последний слой переобучить на целевой задаче для перенастройки весов.

*Метод по шагам:*

- Строится граф из входных данных. Есть два варианта построения графов: co-occurrence graph estimation (CoGE) [10] и supervised graph estimation (SGE) [11]. CoGE напрямую определяет близость элементов данных основываясь на частоте смежности. SGE автоматически запоминает признаки сходства элементов.
- Строится лапласиан полученного графа. Определяется операция свертки.
- Применяется сверточная нейронная сеть к графикам. Модель SCNN (спектральная сверточная нейронная сеть) обучается на информации, полученной на предыдущем шаге. Обучение определят веса каждого слоя путем минимизации функции потерь, специфичной для каждой задачи.
- Выделение признаков для переноса.
- Обучение переносом в спектральном домене. Строится модель для целевой задачи копированием сверточных и субдискретизирующих слоев, которые содержат признаки обученные для исходного домена и исходной задачи. А затем, последний, полно связанный слой модели переобучается на новом, целевом домене для настройки весов.

*Результаты:* передача знаний между доменами более эффективна, когда представление графа в исходном и целевом домене более схожи.

*Будущее развитие:* применить подход к различным наборам данных в разных областях.

## SR2LR [3]

*Задача:* предсказание связей между вершинами.

*Подход:* Inductive transfer learning. Relational Knowledge transfer.

*Основная идея:*

Данные в исходном и целевом доменах описываются марковскими логическими сетями. Предикаты описывают отношения между сущностями. Предполагается, что хорошая модель исходного домена содержит дизъюнкты двух типов - короткодействующие (short-range clauses), которые касаются свойств единственной сущности и дальнодействующие (long-range clauses), относящийся к свойствам нескольких сущностей. Возможные отображения дизъюнктов первого типа в целевой домен могут быть вычислены на доступных целевых данных непосредственно. Поэтому метод основан на том, чтобы использовать короткодействующие дизъюнкты с целью найти отображения между отношениями в двух доменах, а затем эти отображения использовать для передачи дальнодействующих дизъюнктов. Отсюда и название метода SR2LR - short-range to long-range.

*Сравнение с другими методами:*

- Одна из областей, где обучение с переносом особенно эффективно работает - это статистическое относительное обучение (statistical relational learning SRL). В SRL обучаются вероятностные модели на основе сложных реляционных данных [4], [5]. Примеры обучения (они называются мега-примеры) обычно огромные, содержат сотни сущностей и связей различной длины. Из-за этого, как правило, алгоритмы SRL имеют длительное время обучения и часто чувствительны к локальным максимумам.
- Для решения проблем точности и скорости обучения и эффективен подход обучения с переносом [6] [7]. Но эти подходы требуют большого количества данных, как минимум один целевой мега-пример, состоящий из большого количества сущностей. А метод SR2LR справляется с проблемой и решает задачу в случае немалого количества сущностей, в экстремальном варианте - всего с одним.

- Structure-mapping engine (SME) [8] тоже переносит исходные знания в целевой домен. Метод производит отображение предикатов на основе синтаксического структурного критерия, называемого системностью, и не учитывает точность результирующих выводов в целевых данных. В отличие от SR2LR, который оценивает отображения на основе того, производят ли они эмпирически адекватные дизъюнкты в целевой области.

*Будущее развитие:* новые способы отображения исходных знаний, такие как, отображение предикатов разной арности друг в друга, отображение конъюнкций двух предикатов в исходном домене в один предикат в целевом домене и наоборот.

## AdaGCN [12]

Алгоритмы доменной адаптации (domain adaptation) справляются с задачами обучения с переносом в ситуациях, когда распределение вероятностей в исходном и целевом домене различаются.

*Задача:* Классификация узлов. Использование частично помеченной исходной сети для того, чтобы облегчить классификацию узлов в другой, полностью не помеченной сети.

*Подход:* Transductive transfer learning, feauture representation transfer.

*Основная идея:* метод состоит из двух компонент - компонента полуконтролируемого обучения (найти границу классов с помощью сверточных сетей) и компонента состязательной доменной адаптации (найти представления вершин, инвариантные относительно доменов с помощью состязательного обучения)

1 компонента: представляем  $k$ -ый сверточный слой в исправленном виде относительно обычного GCN:

$$\mathbf{H}_g^k = \sigma \left( \hat{\mathbf{A}}^{n_I} \mathbf{H}_g^{k-1} \mathbf{W}_g^k \right), \quad (1)$$

где  $\hat{\mathbf{A}}$  – сумма матрицы смежности графа и диагональной матрицы степеней вершин графа,  $H_g^k$  –  $k$ -ый сверточный слой,  $W_g^k$  – матрица весов,  $n_I$  – параметр сглаживания. С помощью установления подходящего данного параметра, можно контролировать силу сглаживания свертки графа для облегчения переноса знаний и классификации, избегая переобучения.

Далее, после представления вершин исходного и целевого домена в виде эмбедингов, эти данные передаются в классификатор для прогноз-

зирования меток. Классификатор может быть и однослойными классификатором логистической регрессии и многослойным перцептроном.

2 компонента: доменная адаптация моделируется как игра двух игроков, так же как и в GAN. Сеть для генерации эмбедингов вершин играет роль генератора, а играющий роль дискриминатора доменный критик оптимизируется для задачи классификаций эмбедингов вершин на два класса: вершины из исходной сети и вершины из целевой.

После состязательного обучения, могут быть получены инвариантные представления сети и информация о классах может быть передана из исходного в целевой домен. Итого, для обучения инвариантных узлов решается следующая minimax проблема:

$$\min_{\theta_g} \max_{\theta_d} \{\mathcal{L}_d - \gamma \mathcal{L}_{grad}\}, \quad (2)$$

где  $\theta_g, \theta_d$  параметры обучения модели,  $\mathcal{L}_d$  функция потерь доменного критика по отношению к  $\theta_d$ , а  $\mathcal{L}_{grad}$  это штрафной градиент для параметра  $\theta_d$ ,  $\gamma$  это штрафной коэффициент, который должен быть установлен нулем для генератора.

*Сравнение с другими методами:*

- Существующие эмбединговые методы [13], [14], [15] сначала изучают компактные представления узлов для сохранения информации о структуре сети, а затем обучают классификатор с помощью изученных представлений для классификации узлов. Такие методы не позволяют решить следующие проблемы:
  - значительное расхождение между исходным и целевым доменом и отсутствие общих свойств.
  - отсутствие межсетевых ребер для распространения знаний из исходной в целевую сеть.
  - если в исходном домене помечена только малая часть вершин.
- Методы полуконтролируемого обучения основанные на графах [16], [17] показали высокую эффективность для классификации вершин на единственной сети даже в случае когда помеченных вершин не очень много. Методы GCN [17], GraphSAGE [18], GAT [19] показали высокую производительность при классификации вершин, посредством интеграции топологии графов, свойств вершин и наблюдаемых меток вершин в сквозную среду обучения. Но эти методы построены для обучения задания в единственном домене и имеют

проблемы в обобщении на другие домены, которые могут иметь существенно отличающийся набор атрибутов.

- Несколько методов [20], [21] предложены для использования оношений между сетями для повышения эффективности. Они обучают эмбеддинги для нескольких сетей одновременно, но они сильно зависят от существования межсетевых соединений. Как уже было сказано выше, они не справляются с ситуацией отсутствия этих связей.
- Обычный метод доменной адаптации использует знания исходных доменов для того, чтобы решить такую же задачу в целевом домене [22], [23]. Практически все методы в основном ориентированы на векторные данные, такие как картинки и текст, а не графовые структуры данных.

*Будущее развитие:* Исследование передачи знаний из множества исходных сетей в одну целевую сеть и изучение условной адаптации состоятельного домена для лучшего согласования распределения мультимодальных данных.

## AS-MAML [24]

Вариант переноса с обучением, когда требуемый результат получают основываясь только на нескольких/одном тренировочной примере соответственно. Этот тип крайне важен в реальном мире, когда невозможно для каждой задачи собрать большое количество помеченных тренировочных данных для каждого класса и в случаях, когда классы постоянно добавляются.

*Подход:* Inductive. Parameter transfer.

*Задача:* Классификация графов.

Дан граф  $G = (G_1, y_1), (G_2, y_2), \dots, (G_n, y_n)$ , где  $G_i = (V_i, E_i, X_i)$ . В зависимости от метки  $y_i$  он разбивается на тренировочное и тестовое множество  $\{(G^{train}, y^{train})\}$  и  $\{(G^{test}, y^{test})\}$ . При этом  $y^{train}, y^{test}$  не должны иметь общих классов. На этапе обучения на каждом шаге выбирается задача  $\mathcal{T}$  и каждая задача содержит поддерживающие данные  $\mathcal{D}_{sup} = \{(G_i^{train}, \mathbf{y}_i^{train})\}_{i=1}^s$  и данные запроса  $\mathcal{D}_{que} = \{(G_i^{test}, \mathbf{y}_i^{test})\}_{i=1}^q$ . Имея помеченные данные в  $\mathcal{D}_{sup}$ , необходимо предсказать метки в  $\mathcal{D}_{que}$ .

*Основная идея:* Передача знаний от обученной задачи классификации графов к новым задачам. Этот метод является объединением методов GNN [2] в качестве основы и MAML [25] для автоматического подбора оптимальных гиперпараметров.

Этап 1: применение GNN. Цель GNN это представить граф в виде эмбедингов. Это делается в предположении теоремы Банаха о неподвижной точке, так как каждый эмбединг зависит от эмбедингов своих соседей. Существуют различные методы GNN, в зависимости от того в каком виде представлена функция для построения эмбединга. В методе AS-MAML используется агрегатор среднего значения, как в методе GraphSAGE [18]:

$$\mathbf{h}_v^l = \sigma \left( \mathbf{W} \cdot \text{mean}(\{\mathbf{h}_v^{l-1}\} \cup \{\mathbf{h}_u^{l-1}, \forall u \in \mathcal{N}(v)\}) \right), \quad (3)$$

где  $\mathbf{h}_v^l$  - это  $l$ -ый слой представления вершины  $v$ ,  $\sigma$  - сигмоидная функция,  $\mathbf{W}$  параметры агрегатора,  $\mathcal{N}(v)$  содержит соседей вершины  $v$ .

А затем использовать обычную сверточную нейронную сеть.

Этап 2: применение MAML для реализации быстрой адаптации. [26] предложил основанный на обучении с подкреплением (Reinforcement Learning - RL) шаговый контроллер для проведения метаобучения для предсказания связей между вершинами. Но выбранная функция потерь является слишком неоптимальной, чтобы рассматриваться как вознаграждение за преодоление переобучения. В связи с этим и разработан новый шаговый контроллер для ускорения обучения и преодоления переобучения. Он также управляет RL, но оптимальный шаг адаптации выбирается исходя из показателя ANI (Average Node Information) и функции потерь в качестве входных данных и точности классификации в качестве вознаграждения:

*Сравнение с другими методами:*

- Существует несколько методов классификации вершин [27], [28], [29], [30], основанные на GNN и ускоряющие передачу знаний и методы предсказания связей [26] между вершинами. Но они не могут быть расширены на задачи классификации графов.
- В [31] предложена классификация графов, основанная на спектральных измерениях, но этот метод предполагает, что тестовые классы принадлежат тому же множеству классов, что и тренировочные данные.

*Будущее развитие:* Данный метод может быть расширен на другие задачи классификации графов, как например распознавание действий по скелету и анализ подграфов для социальных сетей.

## Выводы

Было рассмотрено несколько различных методов, использующих совершенно разные подходы: и Inductive и Trasductive обучение, перенос и параметров, и представлений признаков, и относительных знаний. Методы, решающие задачи классификации узлов и графов и задачи предсказания связей между вершинами. Подходы в данных методах отличаются значительно: одни методы достаточно просты и лишь на последнем слое нейронной сети меняют веса, другие состоят из нескольких компонент, каждая из которых решает отдельную задачу. Метод SR2LR стоит особняком, так как рассматривает графы знаний, отличающиеся представлением зависимостей между вершинами графа.

Основное будущее развитие всех методов это применение их к различным наборам данных и другим задачам, а также улучшения методов, направленные на уменьшение используемых данных, необходимых для обучения, на разработки других способов отображения данных из исходного домена в целевой и тд.

## Список использованной литературы

### Список литературы

- [1] S. Pan, Q. Yang. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10):1345–1359.
- [2] Z. Wu, S. Pan, F. Chen, G. Long,C. Zhang, S. Yu Philip. 2019. A comprehensive survey on graph neural networks. arXiv:1901.00596.
- [3] Mihalkova, Lilyana and Mooney, Raymond J. 2009. Transfer learning from minimal target data by mapping across relational domains. *Twenty-First International Joint Conference on Artificial Intelligence*.
- [4] L. Getoor, B. Taskar, editors. Introduction to Statistical Relational Learning. MIT Press, Cambridge, MA, 2007.
- [5] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.
- [6] L. Mihalkova, T. Huynh, and R. J. Mooney. Mapping and revising Markov logic networks for transfer learning. (AAAI-07).
- [7] J. Davis, P. Domingos. Deep transfer via second-order markov logic. In *Proceedings of the AAAI Workshop on Transfer Learning For Complex Tasks*. 2008
- [8] Kenneth D. Forbus , Dan Oblinger. Making SME greedy and pragmatic. (CogSci-90).
- [9] J. Lee, H. Kim, J. Lee, and S. Yoon, “Transfer learning for deep learning on graph-structured data.” in AAAI, 2017, pp. 2154–2160
- [10] Sonawane, S., and Kulkarni, P. 2014. Graph based representation and analysis of text document: A survey of techniques. *International Journal of Computer Applications* 96(19).
- [11] Henaff, M.; Bruna, J.; and LeCun, Y. 2015. Deep convolutional networks on graph - structured data. arXiv:1506.05163.
- [12] Quanyu Dai, Xiao Shen, Xiao-Ming Wu, Dan Wang. 2019. Network Transfer Learning via Adversarial Domain Adaptation with Graph Convolution. arXiv:1909.01541(2019)

- [13] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: online learning of social representations,” in KDD, 2014, pp. 701–710.
- [14] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, “LINE: large-scale information network embedding,” in WWW, 2015, pp. 1067–1077.
- [15] A. Grover and J. Leskovec, “node2vec: Scalable feature learning for networks,” in KDD, 2016, pp. 855–864.
- [16] Z. Yang, W. W. Cohen, and R. Salakhutdinov, “Revisiting semi-supervised learning with graph embeddings,” in ICML, 2016, pp. 40–48.
- [17] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in ICLR, 2017.
- [18] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” in NIPS, 2017.
- [19] [12] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Li‘o, and Y. Bengio, “Graph attention networks,” CoRR, 2017.
- [20] J. Ni, S. Chang, X. Liu, W. Cheng, H. Chen, D. Xu, and X. Zhang, “Co-regularized deep multi-network embedding,” in WWW, 2018, pp. 469–478.
- [21] L. Xu, X. Wei, J. Cao, and P. S. Yu, “Embedding of embedding (EOE): joint embedding for coupled heterogeneous networks,” in WSDM, 2017, pp. 741–749.
- [22] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [23] M. Wang and W. Deng, “Deep visual domain adaptation: A survey,” *Neurocomputing*, vol. 312, pp. 135–153, 2018
- [24] Ma, Ning, et al. "Few-Shot Graph Classification with Model Agnostic Meta-Learning."arXiv preprint arXiv:2003.08246 (2020).
- [25] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." *Proceedings of the 34th International Conference on Machine Learning- Volume 70*. JMLR. org, 2017

- [26] Zhengxiao Du, Xiaowei Wang, Hongxia Yang, Jingren Zhou, and Jie Tang. Sequential scenario-specific meta learner for online recommendation. In KDD’19, page 2895–2904, 2019.
- [27] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In ICLR, 2018.
- [28] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D. Yoo. Edge-labeling graph neural network for few-shot learning. In CVPR, June 2019.
- [29] Lu Liu, Tianyi Zhou, Guodong Long, Jing Jiang, and Chengqi Zhang. Learning to propagate for graph meta-learning. In NeurIPS, 2019.
- [30] Huaxiu Yao, Xian Wu, Zhiqiang Tao, Yaliang Li, Bolin Ding, Ruirui Li, and Zhenhui Li. Automated relational meta-learning. In ICLR, 2020.
- [31] Jatin Chauhan, Deepak Nathani, and Manohar Kaul. Few-shot learning on graphs via super-classes based on graph spectral measures. In ICLR, 2020.