Санкт–Петербургский государственный университет

Куликов Михаил Алексеевич

Отчет по учебной практике

Прогнозирование коллабораций в сети соавторств

Научный руководитель:
доцент кафедры ИАС, к. ф.-м. н. Н. Г. Графеева

Санкт-Петербург
2021 г.

Содержание

Введение

За последние несколько лет в мире резко возросло количество доступных конференций и журнальных статей. Для того чтобы ознакомиться со всеми соответствующими последними достижениями, исследователю потребуется просмотреть сотни статей, что отнимает много времени. Как следствие, поиск соавторов со схожими интересами может оказаться сложной задачей. Компании заказчики также испытывают огромные проблемы с набором научных коллективов.

Решение данной проблемы лежит в построении рекомендательной системы, которая могла бы предлагать соавторов. Эту задачу можно сформулировать как проблему прогнозирования связи (ПС) [1], где модель должна предсказать вероятность появления ребра между каждой парой узлов в графе.

Постановка задачи

Целью этой работы является построение модели для прогнозирования вероятности коллаборации между авторами научных статей отраслевого журнала «Нефтяное хозяйство»[20]. Для достижения цели были поставлены следующие задачи:

1. Провести обзор существующих решений. По результатам обосновать выбор того или иного существующего решения или представить собственное.
2. Проанализировать статьи данного журнала.
3. Выбрать метрики для оценки качества прогнозирования.
4. Реализовать выбранное решение.
5. Сравнить качество прогнозирования, используя выбранные метрики, на исследуемом наборе статей.

Глава 1. Обзор

1.1 Эвристические топологические методы

Значительная часть методов прогнозирования связей основывается на некоторой эвристике, которая по структуре графа подсчитывает некоторую метрику схожести между узлами. Наиболее популярные из данного класса методов:

- Common Neighbours [3]
- Adamic-Adar [2]
- Jaccard Coefficient [4]
- SimRank [8]

Одним из главных минусов этих методов является то, что они учитывают только топологию графа, что зачастую бывает недостаточно в анализе социальных сетей, где у каждого узла есть дополнительная информация. Основными плюсами является интерпретируемость и независимость от размера графа. Эмпирические сравнения этих эвристик в различных сетях можно найти в [6, 7]. В таблице 1 приведены четыре популярных эвристики, которые будут использоваться в дальнейшем.

Название	Формула
Common Neighbours	$\lvert\Gamma(x) \cap \Gamma(y)\rvert$
Adamic-Adar	$\sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log\lvert\Gamma(z)\rvert}$
Jaccard Coefficient	$\frac{\lvert\Gamma(x) \cap \Gamma(y)\rvert}{\lvert\Gamma(x) \cup \Gamma(y)\rvert}$
SimRank	$score(x,y) = C\frac{\sum_{a \in \Gamma(x)} \sum_{b \in \Gamma(y)} score(a,b)}{\lvert\Gamma(x)\rvert\lvert\Gamma(y)\rvert}$

Таблица 1: Популярные эвристики для прогнозирования связи. x, y — узлы. $C = \text{const}$. $\Gamma(x)$ — множество вершин смежных с вершиной x.

1.2 Графовые нейронные сети

Графовая нейронная сеть (ГНН) — это новый тип нейронной сети для обучения по графам [9]. В основе ГНН заложен механизм распространения информации. Граф обрабатывается набором модулей, которые связаны между собой в соответствии со связями графа. Также каждый из модулей связан с узлами графа. В процессе обучения, модули обновляют свои состояния и обмениваются информацией. Это продолжается до тех пор, пока модули не достигнут устойчивого равновесия (для того, чтобы была гарантия того, что такое устойчивое состояние существует, этот механизм распространения ограничен). Выходные данные ГНН вычисляются на основе состояния модуля на каждом узле.

Было эмпирически выявлено, что ГНН отлично себя показывают в задачах, связанных с анализом социальных сетей [10]. Так как создание новой архитектуры ГНН строго под каждую задачу достаточно затруднительно, то самым разумным вариантом является взять архитектуру ГНН, которая хорошо себя показала на различных наборах данных и позволяет прогнозировать связь, используя не только топологические признаки, но и явные признаки узла, например векторное представление темы статьи. Данным требованиям удовлетворяет фреймворк SEAL [10].

Основным плюсом ГНН является то, что при прогнозировании учитываются не только топология графа, но и явные признаки узла. К сожалению, ГНН неинтерпретируемы, а также хорошие результаты прогнозирования были достигнуты на довольно больших графах, что не гарантирует хороших результатов при решении данной задачи.

1.3 Тематическое моделирование

Тематическое моделирование — это одно из направлений обработки естественного языка. Тематическая модель коллекции текстовых документов определяет: к каким темам относится каждый документ, и какие слова образуют каждую тему.

Вероятностная тематическая модель описывает каждую тему дискретным распределением вероятностей слов, а каждый документ — дискретным распределением вероятностей тем. Тематическая модель преобразует любой текст в вектор вероятностей тем. Особенность вероятностных тематических векторных представлений текста в том, что они интерпретируемые.

Тематическое моделирование похоже также на кластеризацию документов. Отличие в том, что при кластеризации документ целиком относится к одному кластеру, тогда как тематическая модель осуществляет мягкую кластеризацию, разделяя документ между несколькими темами.

Существует несколько наиболее популярных методов вероятностного тематического моделирования:

- PLSA [11]
- LDA [12]
- ARTM [13]
- hARTM [13]

Каждой из этих моделей можно воспользоваться с помощью библиотеки BigARTM [14]. Имея векторное представление каждого из документов, то есть статей в нашем случае, становится возможным определить новую метрику схожести авторов, основанную на косинусной близости [15].

$$SIM_{cosin}(u,v) = S(P_u, P_v) \times \frac{1}{|\Gamma(u) \cap \Gamma(v)|} \times \sum_{z \in \Gamma(u) \cap \Gamma(v)} S(P_{uz}, P_{vz})$$

В этом уравнении $S(P_u, P_v)$ является степенью схожести между двумя множествами статей, написанных авторами u и v. $S(P_{uz}, P_{vz})$ является степенью схожести между двумя множествами статей, написанных парами авторов (u, z) и (v, z). Обозначим множество векторов, соответствующих множествам статей P_u, P_v, P_{uz}, P_{vz}, как X_u, X_v, X_{uz}, X_{vz}, где $X_u = \{x_{u1}, \ldots, x_{um}\}$, $X_v = \{x_{v1}, \ldots, x_{vn}\}$, $X_uz = \{x_{uz1}, \ldots, x_{uzk}\}$, $X_vz = \{x_{vz1}, \ldots, x_{vzq}\}$. Определим степень схожести между двумя множествами статей следующим образом:

$$S(P_u, P_v) = \frac{1}{e^{1-cosin(x_u^{av}, x_v^{av})}} \text{ , где } x_u^{av}(j) = \frac{1}{m}\sum_{i=1}^{m} x_{ui}(j),\ j = \overline{1, K}$$

1.4 Бинарная классификация

Прогнозирование связи в сети коллабораций возможно сформулировать как проблему бинарной классификации, то есть пара авторов может быть классифицирована как положительный класс или как отрицательный. Если пара авторов принадлежит положительному классу, то они будут иметь коллаборацию в будущем. Если пара принадлежит отрицательному, то коллаборации не будет. В таком случае мы можем использовать как топологические эвристические признаки, так и основанные на тематическом моделировании признаки, подав их на вход нашей модели бинарной классификации.

Глава 2. Статьи для исследования

В качестве статей для исследования были выбраны статьи, опубликованные в журнале «Нефтяное хозяйство» [16] с 2012 по 2017 год. Основными плюсами данного журнала являются наличие подготовленного набора данных, который включает в себя авторов статьи, текст статьи, аффилиацию авторов.

В следующей таблице указаны основные параметры данного набора статей:

Параметр	Значение
Число авторов	3517
Число публикаций	1459
Число связей	10890
Число уникальных аффилиаций	148

Таблица 2: Параметры набора статей

Поскольку использование вероятностного тематического моделирования требует от корпуса текстов выполнение некоторых гипотез [13], то тексты статей были предобработаны, используя стандартные методы работы со строками. Для приведения всех слов в нормальную форму использовалась библиотека pymorphy2 [18], не имеющая аналогов. Далее используя PLSA, LDA, ARTM, а также hARTM были построены тематические модели данного корпуса текстов. Качество моделей проверялось эмпирически с помощью библиотеки pyLDAvis [17], являющейся единственной поддерживаемой на данный момент библиотекой для визуализации тематических моделей на Python. Наилучшие результаты с точки зрения интерпретируемости тем показала hARTM. Было выделено 25 тем. Тем самым было получено векторное представление для каждой из статей, что будет в дальнейшем использовано для вычисления метрики схожести SIM_{cosin}.

Глава 3. Формирование набора данных

Необходимо сформировать набор признаков, который будут подаваться на вход бинарному классификатору.

- Топологические признаки: Common Neighbours, Adamic-Adar, Jaccard Coefficient, SimRank.
- Признак, основанный на тематике написанных авторами статей — SIM_{cosin}.
- Признак, основанный на аффилиации авторов.

Наиболее важным компонентом является формулирование целевой переменной так, чтобы не было утечек, а также сохранялась возможность разделения набора данных на части для скользящего контроля. Было решено разделить набор данных на две части: в первую часть попадают статьи, написанные с 2012 по 2015 включительно, во вторую часть попадают статьи, написанные с 2016 по 2017 включительно. После этого рассматриваются статьи только тех авторов, которые писали в обоих временных интервалах. Выбор подобных временных интервалов обусловлен тем, что в таком случае размер набора рассматриваемых статей будет максимальным.

Далее все признаки, требуемые для классификатора, подсчитываются, используя только первую часть набора данных. Целевой переменной будет наличие или отсутствие коллаборации между авторами во втором временном интервале. Таким образом, утечка была избежена, а также подобная постановка задачи позволяет нам свободно делить набор данных на части для скользящего контроля.

Схематически это изображено на рисунке 1.

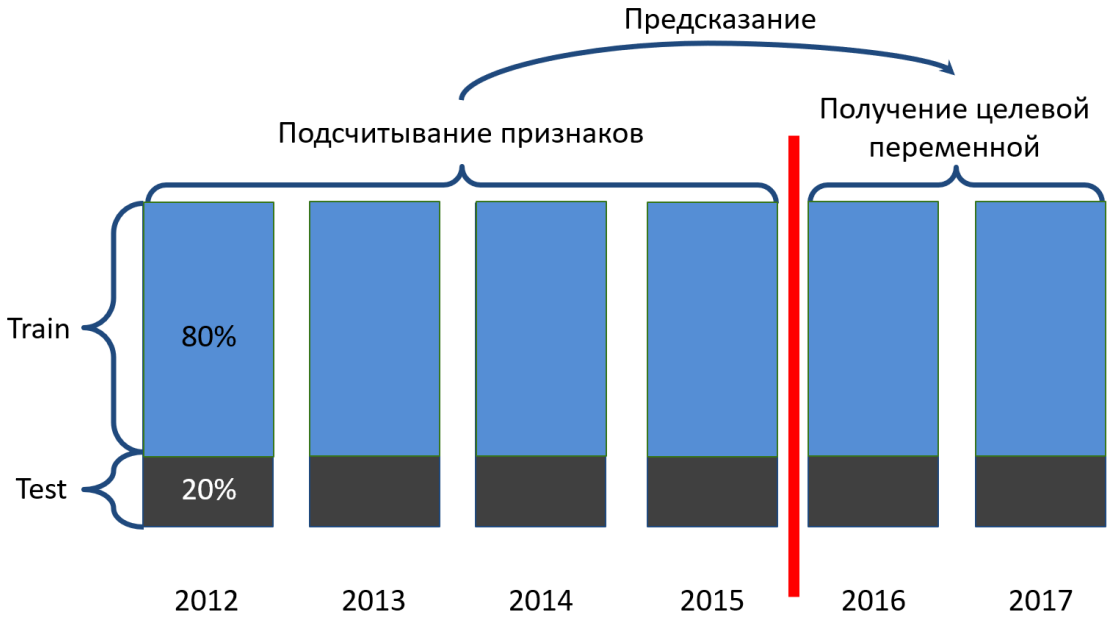

Рис. 1: Схема скользящего контроля.

Глава 4. Выбор метрики оценки качества

После того как задача была сформулирована как задача бинарной классификации, необходимо выбрать метрику оценки качества исследуемых моделей бинарной классификации. Нужно быть крайне аккуратным поскольку неправильный выбор метрики может привести к некорректным выводам о качестве решения.

Так как к положительному классу были отнесены только те пары авторов, которые коллаборировали в интервале с 2016 по 2017 включительно, а к отрицательному классу отнесены все остальные пары, набор данных получается крайне несбалансированным. Элементов положительного класса всего 0.6% от общего числа элементов в наборе данных. Следовательно нельзя использовать метрики, чувствительные к несбалансированному набору данных.

Важную роль в оценке качества любой модели машинного обучения играют точность (precision) и полнота (recall) [19]. Точность в нашей задаче можно интерпретировать как долю предсказанных моделью коллабораций, реализовавшихся с 2016 по 2017 год включительно. Полноту, как долю предсказанных коллабораций от всех, реализовавшихся в интервале с 2016 по 2017 год включительно. Высокие показатели обеих метрик важны для данной задачи. Но поскольку временной интервал, на котором измеряется качество предсказаний, сильно ограничен, то часть связей, предсказанных моделью, могли не успеть реализоваться в данном интервале, поэтому точность в данной задаче играет гораздо меньшую роль нежели полнота предсказания. Сравнение моделей по какой-то одной из этих метрик некорректно. Модель с высокой точностью может иметь низкую полноту, что на практике означает, что она могла отнести правильно лишь один элемент к положительному классу, а остальные элементы отнести к отрицательному классу. Модель с высокой полнотой и низкой точностью могла отнести весь набор данных к положительному классу. Обе модели на практике будут бесполезны.

Было решено использовать метрику, которая учитывает как точность, так и полноту предсказания, а именно F_β меру, как основную метрику для оценки качества модели. Она определяется следующим образом:

$$F_\beta = (1 + \beta^2) \times \frac{precision \times recall}{\beta^2 \times precision + recall}$$

В качестве значения β было выбрано 3, так как в таком случае полнота будет учитываться больше чем точность, что идеально подходит для данной задачи.

Глава 5. Обучение моделей

Для обучения были выбраны стандартные модели для решения задачи бинарной классификации, а также ГНН.

- Рандомный лес [24]
- Логистическая регрессия
- SEAL (ГНН)
- Градиентный бустинг [21]
- SVM [22]

CatBoost [20] был выбран в качестве библиотеки для обучения модели градиентного бустинга из-за хорошей документации и возможности вычисления на GPU.

Для обучения ГНН был выбран фреймворк SEAL, преимущества которого были описаны в 1.2.

Для обучения SVM была использована библиотека ThunderSVM [23], чье главное преимущество в поддержке вычислений на GPU, что значительно сокращает время обучения.

Глава 6. Результаты

Наилучшие результаты модели показали при использовании всех признаков, упомянутых ранее. Подсчет метрик производился на тестовом наборе данных. На наборе данных для обучения производился подбор гиперпараметров с помощью метода скользящего контроля.

Модель	Precision	Recall	ROCAUC	PRAUC	F3
Лог. регрессия	0.378	0.646	0.97	0.5	0.6
SVM	0.33	0.65	0.97	0.52	0.595
Град. бустинг	0.357	0.63	0.968	0.54	0.586
Рандомный лес	0.44	0.53	0.94	0.507	0.52
SEAL (ГНН)	0.24	0.34	0.75	0.19	0.33

Таблица 3: Полученные результаты на тестовом наборе данных для каждой из моделей с подобранными гиперпараметрами.

Наилучший результат показала логистическая регрессия.

Собранный набор данных, с которым проводилась работа размещен в репозитории GitHub [25].

Заключение

В данной работе были достигнуты следующие результаты.

- Проведен анализ существующих решений и методик прогнозирования связей.
- Проанализированы статьи отраслевого журнала «Нефтяное хозяйство».
- Выбрана метрика для оценки качества прогнозирования.
- Обучен ряд моделей для решения задачи бинарной классификации.
- Используя выбранную метрику, была отобрана лучшая модель.

Список литературы

[1] Li X., Chen H. Recommendation as link prediction: a graph kernel-based machine learning approach //Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries. – 2009. – С. 213-216.

[2] Adamic L. A., Adar E. Friends and neighbors on the web //Social networks. – 2003. – Т. 25. – №. 3. – С. 211-230.

[3] Newman M. E. J. Clustering and preferential attachment in growing networks //Physical review E. – 2001. – Т. 64. – №. 2. – С. 025102.

[4] Chowdhury G. G. Introduction to modern information retrieval. – Facet publishing, 2010.

[5] Zhou T., Lü L., Zhang Y. C. Predicting missing links via local information //The European Physical Journal B. – 2009. – Т. 71. – №. 4. – С. 623-630.

[6] Lü L., Zhou T. Link prediction in complex networks: A survey //Physica A: statistical mechanics and its applications. – 2011. – Т. 390. – №. 6. – С. 1150-1170.

[7] Klein D. J., Randić M. Resistance distance //Journal of mathematical chemistry. – 1993. – Т. 12. – №. 1. – С. 81-95.

[8] Jeh G., Widom J. Simrank: a measure of structural-context similarity //Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. – 2002. – С. 538-543.

[9] Bruna J. et al. Spectral networks and locally connected networks on graphs //arXiv preprint arXiv:1312.6203. – 2013.

[10] Zhang M., Chen Y. Link prediction based on graph neural networks //arXiv preprint arXiv:1802.09691. – 2018.

[11] Hofmann T. Probabilistic latent semantic indexing //Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. – 1999. – С. 50-57.

[12] Blei D. M., Ng A. Y., Jordan M. I. Latent dirichlet allocation //the Journal of machine Learning research. – 2003. – Т. 3. – С. 993-1022.

[13] Vorontsov K. V. Additive regularization for topic models of text collections //Doklady Mathematics. – Pleiades Publishing, 2014. – Т. 89. – №. 3. – С. 301-304.

[14] Сайт библиотеки BigARTM - https://bigartm.org/

[15] Chuan P. M. et al. Link prediction in co-authorship networks based on hybrid content similarity metric //Applied Intelligence. – 2018. – Т. 48. – №. 8. – С. 2470-2486.

[16] Сайт журнала «Нефтяное хозяйство». — URL: https://oil-industry.net/

[17] Репозиторий библиотеки pyLDAvis. // GitHub — URL: https://github.com/bmabey/pyLDAvis

[18] Korobov M. Morphological analyzer and generator for Russian and Ukrainian languages //International Conference on Analysis of Images, Social Networks and Texts. – Springer, Cham, 2015. – С. 320-332.

[19] Wikipedia. Precision and Recall // Wikipedia, the free encyclopedia. — https://en.wikipedia.org/wiki/Precision_and_recall

[20] Dorogush A. V., Ershov V., Gulin A. CatBoost: gradient boosting with categorical features support //arXiv preprint arXiv:1810.11363. – 2018.

[21] Friedman J. H. Greedy function approximation: a gradient boosting machine //Annals of statistics. – 2001. – С. 1189-1232.

[22] Boser B. E., Guyon I. M., Vapnik V. N. A training algorithm for optimal margin classifiers //Proceedings of the fifth annual workshop on Computational learning theory. – 1992. – С. 144-152.

[23] Wen Z. et al. ThunderSVM: A fast SVM library on GPUs and CPUs //The Journal of Machine Learning Research. – 2018. – Т. 19. – №. 1. – С. 797-801.

[24] Breiman L. Random forests //Machine learning. – 2001. – Т. 45. – №. 1. – С. 5-32.

[25] Репозиторий с набором данных. // GitHub — URL: https://github.com/MihailKulikov/LinkPrediction