УДК 004.891

**Векторные представления рёбер графа онтологии как инструмент для анализа и генерации новых данных.**

**О.П. Мосалов**, кандидат физико-математических наук, заведующий учебно-научной лабораторией интеллектуальных систем управления, Государственное бюджетное образовательное учреждение высшего образования Московской области «Технологический университет имени дважды Героя Советского Союза, лётчика-космонавта А.А. Леонова», г. Королёв, Московская область.

*Рассмотрено несколько способов формирования векторных представлений рёбер в графе онтологии. Проведены вычислительные эксперименты по предсказанию новых рёбер с использованием этих способов. Предложены две метрики для сравнения алгоритмов генерации новых рёбер. Продемонстрирован алгоритм, который может использоваться в качестве базового в дальнейших исследованиях.*

Графы онтологии, векторные представления графов, предсказание рёбер в графах, машинное обучение.

**Edge embedding of ontology graphs as a tool for analysis and generation of new data.**

**O.P. Mosalov**, Ph.D., head of the educational and scientific laboratory of intellectual control systems, State educational institution of higher education Moscow region “Leonov’s University of technology”, Korolev, Moscow region.

*Several methods of building of edge embedding in ontology graphs are considered. Computational experiments to predict new edges by the proposed methods are fulfilled. Two metrics for edge generation algorithms are suggested. An algorithm that can be used as a baseline is demonstrated.*

Ontology graphs, graph embedding, link prediction, machine learning.

Анализ данных в современной экономике является одним из основных конкурентных преимуществ. Однако, не все данные легко представимы в виде, подходящем для машинного анализа. Поэтому в последнее время активно исследуются подходы, позволяющие работать с плохо структурированными, неудобными для обработки, данными.

Одним из таких форматов являются графы онтологий [3,4]. Онтологией называется [1] объединение четырёх множеств: экземпляров, понятий, атрибутов и отношений. Экземпляры соответствуют объектам или сущностям предметной области, понятия – выделенным группам объектов, атрибуты – всевозможным свойствам объектов, отношения – взаимосвязям между объектами.

В данной работе рассматривается структура графа онтологии, в которой нет атрибутов, количество понятий ограничено при построении, а количества экземпляров и отношений могут быть любыми и определяются только имеющимися данными. Таким образом, граф онтологии может быть записан в виде: , где *V* – множество вершин (объектов), *E* – множество рёбер (отношений), *L* – множество слоёв (понятий).

В работе [2] был рассмотрен алгоритм предсказания существования рёбер в графе на основании информации об уже известных рёбрах с использованием генеративно-состязательной сети. Для построения набора образцов, необходимых для настройки такой сети, использовался следующий базовый алгоритм: на основании множества вершин графа *V* строится множество всех возможных рёбер , для каждого возможного ребра определяется – признак его наличия в множестве *E*:

, (1)

после чего формируется множество *X*, состоящее из трёхэлементных кортежей:

, (2)

где , а *f* – некая функция, преобразующая ребра в числовые векторы. Образцы, в итоге, формировались путём конкатенации заранее определённого количества случайно выбранных элементов множества *X*:

(3)

В данной работе описываются и сравниваются несколько вариантов выбора функции *f*.

**Формальная постановка задачи.** Рассматриваются граф онтологии и алгоритм предсказания существования рёбер в этом графе , где и , то есть данный алгоритм позволяет находить новые, ранее неизвестные, отношения между экземплярами онтологии.

Алгоритм имеет внутреннюю структуру: , где преобразует граф *G* в набор образцов заданного формата *S*; генерирует новые данные того же формата; преобразует полученные новые данные обратно к виду ребер графа *G*:

, (4)

, (5)

, (6)

где – множества числовых векторов.

В рамках задачи необходимо построить пару преобразований и , которые, при заданных алгоритме генерации с использованием генеративно-состязательной сети [2], сформируют достаточно качественный результат.

Для этого нужно выбрать формулу *f*, то есть определить правило, по которому ребру графа можно поставить в соответствие числовой вектор: , где – размерность такого вектора.

Преобразование такого объекта, как ребро графа, в числовой вектор называется построением векторного представления или эмбеддинга данного объекта.

**Векторные представления графов.** Векторные представления объектов в машинном обучении являются распространённым способом преобразования информации, которая изначально собрана в виде, отличном от числовых векторов. Наиболее широко данный подход применяется в обработке естественных языков для преобразования текста в числовой формат, удобный для дальнейшего применения существующих алгоритмов и моделей [9].

Успешное применение векторных представлений в задачах, связанных с данными в виде текстов на естественных языках, привело к разработке алгоритмов для построения таких представлений и для других типов сложных объектов, в том числе графов [10]. Аналогично тому, как векторные представления в обработке естественных языков строится изначально на уровне отдельных слов, а затем, при необходимости, объединяются в более высокоуровневые векторные представления предложений и целых текстов, для графов векторные представления изначально строятся для отдельных рёбер.

Существуют алгоритмы различной сложности по построению векторных представлений рёбер графа, учитывающие различную имеющуюся информацию о вершинах ребра, о типе связи, о ближайшей окрестности рассматриваемого ребра, а также другую информацию как о графе, так и в целом о предметной области, которую он описывает [5].

**Рассмотренные реализации векторных представлений и критерии их сравнения.** В данном исследовании рассмотрены два основных подхода к формированию векторных представлений рёбер графов. В обоих случаях для имеющегося графа векторные представления строятся в соответствии с (1, 2):

.

Первый подход не учитывает информацию о слоях и формирует векторные представления (4) только с учётом информации о наличии или отсутствии рёбер в графе. В этом случае в качестве значений функции используется порядковый номер вершины: и .

При этом обратное преобразование сгенерированных векторов к рёбрам графа (6) осуществляется следующим образом: вектор разделяется на частей, каждая из которых имеет вид . Вычисляется среднее количество известных рёбер в образцах и отбирается такое же количество частей с максимальными значениями . Далее вычисляется, каким значениям *i* и *j* соответствуют значения и и получившиеся рёбра считаются сгенерированными.

Второй подход явно учитывает информацию о слоях и формирует векторные представления (4) с учётом принадлежности отдельных вершин ребра к слоям. В таком случае значение в свою очередь является вектором, размерность которого равна количеству слоёв в графе . Тому слою, к которому принадлежит вершина , соответсвует компонента вектора с порядковым номером вершины, остальные компоненты вектора равны нулю:

, где ; .

Таким образом, векторное представление ребра имеет вид .

Для удобства работы с такими векторами их можно сделать плоскими, т.е. преобразовать компоненты вложенных векторов в компоненты самого вектора:

.

Обратное преобразование (6) строится аналогично первому подходу, только и , на основании котррых вычисляются *i* и *j*, являются не числами, а векторами длины . Значения *i* и *j* выбираются как порядковые номера максимальных компонент векторов и соответсвенно.

Сравнивая два описанных подхода, можно отметить, что в первом подходе векторные представления имеют длину , в то время как во втором подходе , то есть вычисление и хранение векторных представлений во втором подходе более затратно. Однако, учёт информации о слоях позволяет лучше описать внутреннее устройство графа онтологии и ожидать более качественные результаты.

Для оценки качества работы отдельных алгоритмов и сравнения их между собой необходима некая метрика. Ранее, в [2] в качестве метрики подобных генеративных моделей рассматривалось расстояние Фреше [7] между распределениями образцов и сгенерированных векторов. Однако, в связи с тем, что при разных способах построения векторных представлений мы имеем дело с пространствами разных размерностей, расстояние Фреше уже не будет удобным инструментом сравнения алгоритмов, хотя для оценки качества работы отдельных алгоритмов его всё ещё можно эффективно использовать.

В качестве более универсальной метрики алгоритмов генерации в данной работе рассматривается количество уникальных сгенерированных новых рёбер *N*, которое определяется следующим образом: после завершения обучения нейросетевого блока алгоритма (5) генерируется большое количество векторов , , из каждого из них извлекаются ранее неизвестные рёбра, после чего вычисляется общее количество уникальных рёбер для всех сгенерированных векторов. Ожидается, что хорошо работающая модель будет приводить к значениям такой метрики, которые, во-первых, не слишком велики по сравнению с количеством известных рёбер, во-вторых, стабилизируются при росте количества сгенерированных векторов:

В качестве дополнительной метрики используется количество уникальных сгенерированных новых рёбер, которые были получены более одного раза. При достаточно больших значениях рёбра, которые были получены ровно в одном экземпляре, могут быть следствием погрешности работы модели либо представлять собой какие-то осмысленные, но требующие дополнительного анализа данные.

Итого, от алгоритма ожидается, что при генерации большого количества векторов, извлечённые из них рёбра будут повторяться, и, в пределе, модель сгенерирует некое множество новых рёбер, размер которого перестанет увеличиваться с ростом количества сгенерированных векторов.

**Описание проведённых вычислительных экспериментов и их основные результаты.** Граф онтологии, исследованный в данной работе, описывает информацию о предметной области, связанной с аддитивными технологиями, 3D-печатью и использованием различных, в том числе неметаллических, материалов. Параметры данного графа следующие: количество вершин – около 350, количество рёбер – около 550, количество слоёв – 4.

Численное моделирование проводилось с использованием языка программирования Python и библиотеки Keras [6]. Исследовались оба описанных выше подхода к построению векторных представлений: с использованием информации о слоях и без неё.

Так как выбор способа построения векторных представлений влияет на их размерность, то значения гиперпараметров генеративно-состязательной сети зависит от анализируемого подхода. Количество элементов множества *X*, используемых для формирования одного образца, во всех экспериментах было выбрано как .

При первом подходе размер векторного представления одного ребра , таким образом размеры входного слоя дискриминатора и выходного слоя генератора равны .

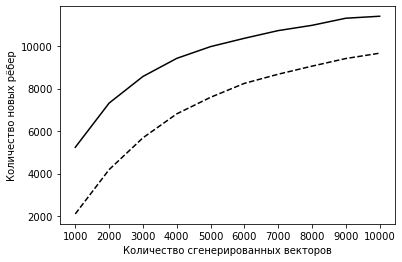
При втором подходе размер векторного представления одного ребра и размеры входного слоя дискриминатора и выходного слоя генератора равны .

Остальные параметры искусственных нейронных сетей генератора и дискриминатора были одинаковы. Генератор имел 100 нейронов во входном слое, 250 – в первом скрытом слое и 1000 – во втором скрытом слое. Но обоих скрытых слоях была использован активационная функция «Leaky ReLU» [8], на выходном слое – линейная функция. Дискриминатор имел 1500 нейронов в первом скрытом слое, 500 – во втором и 50 – в третьем, на всех скрытых слоях также использовалась активационная функция «Leaky ReLU». Выходной слой содержал один нейрон с сигмоидой в качестве активационной функции.

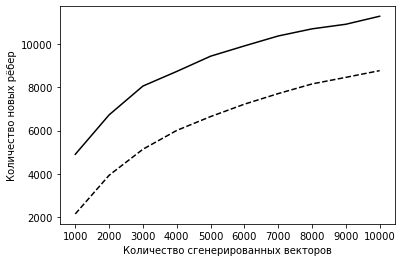
Так же были рассмотрены два способа приведения компонент векторов образцов к единому интервалу значений: линейные преобразования к отрезку [0, 1] или к отрезку [‑1, 1].

Итого, было проведено четыре серии экспериментов: с двумя подходами вычисления векторных представлений и двумя способами приведения к единому интервалу. Для всех четырех вариантов генеративно-состязательных сетей процесс обучения прошёл успешно, однако выбранные метрики показали различные значения.

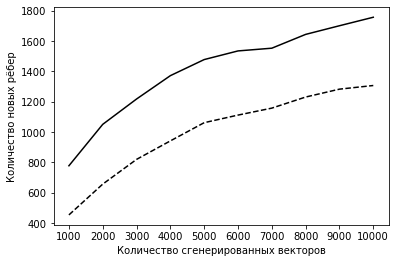
Ниже на рис. 1‑4 представлены графики, показывающие изменение обеих метрик в зависимости от количества сгенерированных векторов. Сплошной линией показано значение количества уникальных сгенерированных новых рёбер, пунктирной линией – количество уникальных сгенерированных новых рёбер, которые были получены более одного раза).



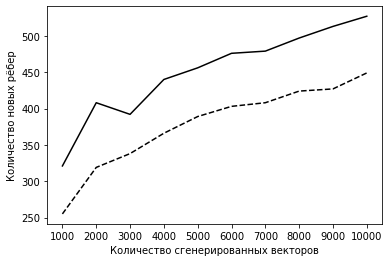
**Рисунок 1 – Векторные представления без учёта информации о слоях, преобразование значений к отрезку [0, 1].**

****

**Рисунок 2 – Векторные представления без учёта информации о слоях, преобразование значений к отрезку [‑1, 1].**

****

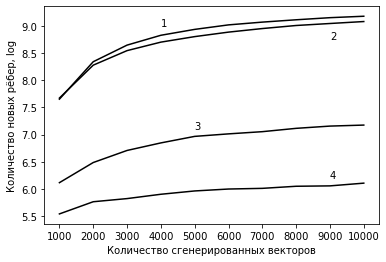
**Рисунок 3 – Векторные представления с учётом информации о слоях, преобразование значений к отрезку [0, 1].**

****

**Рисунок 4 – Векторные представления с учётом информации о слоях, преобразование значений к отрезку [‑1, 1].**

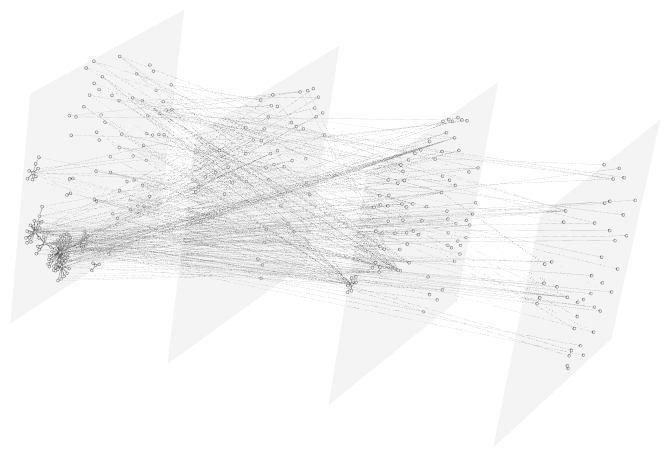
Можно увидеть, что во всех экспериментах наблюдается ожидаемые результаты – рост количества новых уникальных рёбер замедляется с увеличением числа сгенерированных векторов. При этом четвёртый эксперимент показывает наименьшие абсолютные значения новых рёбер, при том, что их количество всё ещё довольно велико, поэтому именно эти результаты имеет смысл рассматривать как базовые, сравнивая другие алгоритмы и наборы гиперпараметров с ним.

Для удобства сравнения на рис. 5 представлены результаты всех четырёх экспериментов. По оси ординат отложены логарифмы величины уникальных сгенерированных новых рёбер, которые были получены более одного раза.

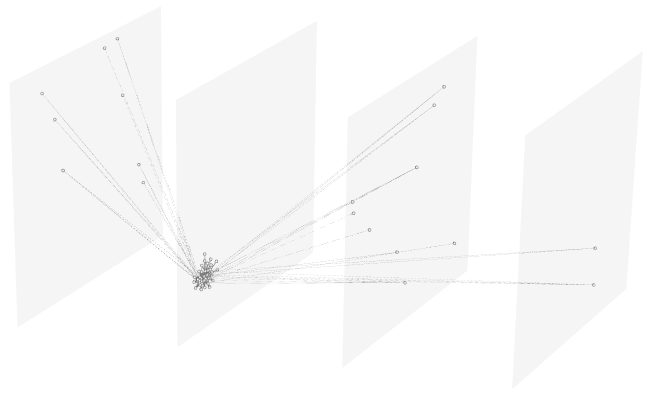


**Рисунок 5 – Сравнение результатов всех четырёх экспериментов (в логарифмической шкале). Цифры соответствуют номерам экспериментов.**

В качестве одного из возможных направлений дальнейшего анализа полученных результатов было проведено сравнение множеств новых рёбер, полученных в рамках третьего и четвёртого экспериментов. Ниже на рис. 6 представлен изначальный граф, а на рис. 7 – рёбра, которые были получены более одного раза и в третьем, и в четвёртом экспериментах.



**Рисунок 6 ­– Изначальный граф.**

****

**Рисунок 7 – Рёбра, сгенерированные по итогам третьего и четвёртого экспериментов.**

**Выводы и планы дальнейших исследований.** В данной работе рассмотрен подход к анализу графов онтологии, связанный с формированием векторных представлений отдельных рёбер графа. Проведены вычислительные эксперименты по предсказанию существования рёбер в графе с несколькими вариантами реализации векторных представлений. Предложен набор гиперпараметров, который может рассматриваться как точка отсчёта (baseline) для других исследований. Сформулированы и применены на практике две метрики генеративных алгоритмов для графов.

В дальнейшем необходимо разработать и рассмотреть другие, более сложные и более точные алгоритмы формирования векторных представлений рёбер графов. При наличии процедуры сравнения различных алгоритмов на базе общих метрик можно будет проводить анализ качества работы отдельных алгоритмов и, при необходимости, эффективно объединять их.

Требуется провести более глубокий анализ множеств, которые получаются при рассмотрении рёбер, полученных более *N* раз при *N* > 1. Возможно введение критерия достоверности для новых рёбер, основанного на том, как часто конкретное ребро получается в результате работы генеративного алгоритма.

Также необходимы дальнейшие исследования рёбер, одновременно предсказанных несколькими различными алгоритмами или различными реализациями одного алгоритма, для того чтобы проверить наличие зависимости между достоверностью ребра и количеством алгоритмов, нашедших его.

*Литература*

1. Лапшин В. А. Онтологии в компьютерных системах // М.: Научный мир, 2010.
2. Мосалов О.П. Использование генеративно-состязательных сетей в задаче предсказания существования рёбер в онтологическом графе // Информационно-технологический вестник – 2020 – № 4(26) – С. 96-103.
3. Муромцев Д., Волчек Д., Романов А. Индустриальные графы знаний – интеллектуальное ядро цифровой экономики // Control Engineering Россия. 2019. № 5 (83). С. 32-39.
4. Орлов Ю.Н., Парфенова Ю.А. Анализ структуры онтологического графа толкового словаря // Препринты ИПМ им. М.В.Келдыша. 2017. № 31. 25 с.
5. Cai H., Zheng V.W., Chen-Chuan Chang K. A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications [Электронный ресурс]. URL: https://arxiv.org/abs/1709.07604 (дата обращения: 02.02.2021).
6. Deep learning for humans [Электронный ресурс]. URL: https://keras.io (дата обращения: 01.06.2020).
7. Fréchet M. Sur la distance de deux lois de probabilité // C. R. Acad. Sci. Paris, 244. 1957. PP. 689–692.
8. Maas A. L., Hannun A. Y., Ng A. Y. Rectifier nonlinearities improve neural network acoustic models // ICML 2013 – 30th International Conference on Machine Learning (ICML), June 16–21, Atlanta, Georgia, Proceedings. 2013.
9. Mikolov T., Sutskever I., Chen K., Corrado G., Dean J. Distributed Representations of Words and Phrases and their Compositionality [Электронный ресурс]. URL: https://arxiv.org/abs/1310.4546 (дата обращения: 05.11.2020).
10. Perozzi B., Al-Rfou R., Skiena S. Deepwalk: Online learning of social representations // In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014. PP. 701–710.