«Московский физико-технический институт
(национальный исследовательский университет)»
Физтех-школа прикладной математики и информатики
Кафедра «Интеллектуальные системы»
Бишук Антон Юрьевич

Контролируемая генерация графов

03.04.01 — Прикладные математика и физика

Выпускная квалификационная работа магистра

Научный руководитель:
к.ф.-м.н.
Зухба Анастасия Викторовна

Москва
2023 г.

Содержание

Аннотация

Данная работа описывает новый метод генерации графов, который использует разделение статистических характеристик графа на две группы. Первая группа, называемая «простыми признаками», может быть вычислена эффективными детерминированными алгоритмами со сложностью не более квадратичной от числа вершин. Вторая группа характеристик генерируется в скрытом пространстве и отвечает за закономерности графа, которые невозможно описать «простыми признаками». Этот подход позволяет генерировать графы с точно заданными статистическими характеристиками, при этом сохраняя их разнообразие. Более того, данный метод может быть применен для генерации графов, имеющих схожую структуру с исходным. Это особенно полезно при работе с графами, описывающими контакты между людьми, например, граф контактов.

1 Введение

Все более популярным становится использование графов в качестве источников данных в задачах машинного обучения. Однако это направление все еще остается недостаточно развитым по причине отсутствия больших наборов данных для обучения и тестирования моделей. Для решения этой проблемы часто используются подходы, основанные на единственном графе, либо на синтетических генераторах графов[1]. Эти методы могут быть неточными или не всегда коррелирующими с реальными данными, которые зависят от конкретной задачи.

Кроме того, часто возникает потребность в графах, имеющих схожее распределение с исходным. Это важно, например, в случае графа контактов фиксированного сообщества, где необходимо сгенерировать ряд графов, похожих на изначальный. Для этой задачи традиционно используются генеративные модели, такие как GraphVAE [2] или диффузионные [3].

Однако существующие методы фокусируются на реконструкции исходного графа, и не способны учитывать при генерации интуитивно понятные характеристики. Например, в случае графа контактов время взаимодействие людей в офисе ограничено, а потому есть ограничения на время и число контактов. Мы же предлагаем использовать в качестве ограничения на генерацию не столько качество реконструкции, сколько заранее выбранные глобальные характеристики графа (например число ребер, вершин, кластерное число и так далее). Тем самым мы можем генерировать графы с заранее выбранными статистиками и имеющие распределение схожее с исходным графом.

Предложенный метод может быть использован для поиска «сложных» статистик графа, таких как центральность смежности вершин[4], поиск которых до сих пор остаётся нерешенным на достаточно высоком уровне. Подробнее об использовании подхода в других задачах будет рассказано в разделе с методом.

В практической части работы приведены эксперименты иллюстрирующие успешную работу нашего метода в генерации графов с заданными статистиками.

2 Обзор литературы

Генерация графов — это активно развивающееся направление в машинном обучении, которое находит применение во многих областях, например, биоинформати-

ке, физике, NLP. Генерация графов может помочь решать различные задачи, такие как поиск наиболее важных узлов в графе, классификация графов, прогнозирование свойств графов. Сейчас существуют разнообразные методы генерации графов, начиная с детерминированных алгоритмов и вариационных автокодировщиков и заканчивая генеративно-состязательными и диффузионными сетями.

2.1 Генеративные модели

Одной из ключевых работ в области глубокого обучения и вероятностного моделирования является статья «Auto-Encoding Variational Bayes» [5], представляющая собой первое упоминание модели вариационных автокодировщиков (VAE).

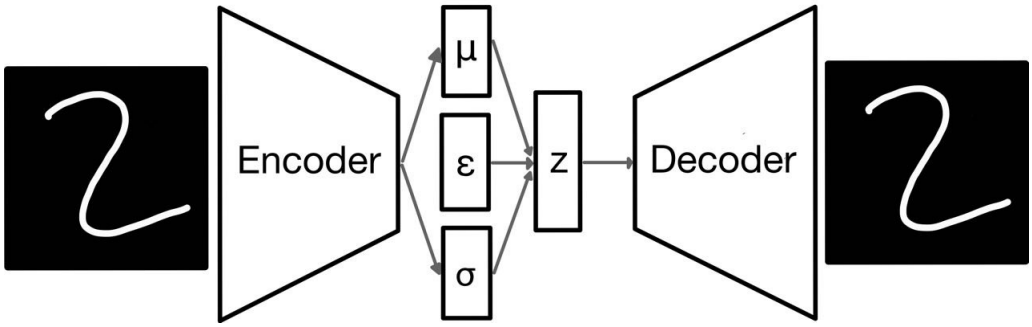

Рис. 1: Схема модели вариационного автокодировщика(VAE). Энкодер задает распределение $p_\varphi(Z|X)$, а декодер $q_\theta(\hat{X}|Z)$, где X и $\hat{X}$ реальный и сгенерированный объекты соответственно; $\varepsilon \in N(0,1)$.

В статье авторы представляют подход к генеративному моделированию данных, который позволяет моделировать сложные распределения и обеспечивает более эффективное обучение в сравнении с классическими методами. Они предлагают использовать нейронную сеть в качестве генеративной модели, которая будет преобразовывать входные данные в скрытое пространство, а затем обратно декодировать из скрытого пространства в исходное, тем самым генерируя объекты исходного пространства. Авторы анонсируют новый подход к уменьшению ошибки для обучения модели, основанный на вариационном выводе. Этот метод позволяет обучать модель вариационным методом, а также получать оценки правдоподобия для сгенерированных данных. Статья является ключевой в развитии вероятностного моделирования в глубоком обучении. Она открыла новые возможности для генеративного моделирования данных, включая генерацию графов при помощи вариационных автокоди-

ровщиков.

Наш метод основывается на использовании дополнительной информации при генерации. Первое упоминание такой идеи встречается в статье «Learning Structured Output Representation using Deep Conditional Generative Models»[6], где, впервые была представлена модель Conditional Variational Autoencoders (CVAE).

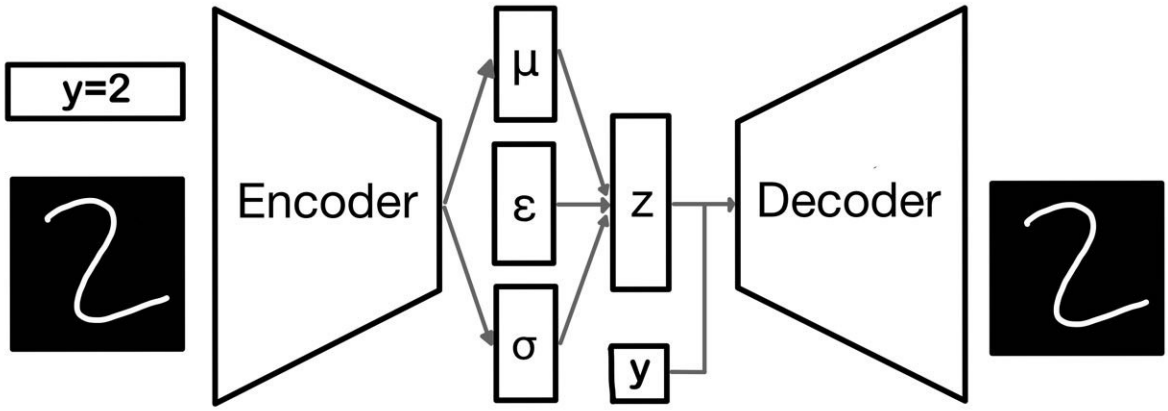

Рис. 2: Схема модели условного вариационного автокодировщика (CVAE). Энкодер задает распределение $p_\varphi(Z|X,y)$, а декодер $q_\theta(\hat{X}|Z,y)$, где X и $\hat{X}$ реальный и сгенерированный объекты соответственно; y – метка объекта; $\varepsilon \in N(0,1)$.

CVAE - это модификация VAE, которая может генерировать данные с заданными условиями. В стандартном VAE модель генерирует данные на основе скрытого пространства, которое не зависит от каких-либо внешних переменных. В CVAE модель использует дополнительную информацию для генерации данных. Авторы в статье показывают, как CVAE может быть использована для генерации изображений с заданными свойствами. Они используют MNIST[7] для генерации цифр с определенными свойствами, такими как цвет и положение цифры на изображении. Также описывается как CVAE может быть использована для классификации изображений. Например, авторы статьи применяют CVAE к задаче классификации CIFAR-10[8], показывая, что CVAE может значительно улучшить точность. Впоследствии было предложено множество модификаций идей CVAE, таких как AC-GAN (Auxiliary Classifier GAN) и InfoGAN (Information Maximizing GAN), которые используют схожие идеи для генерации изображений с более сложными свойствами.

Идея нашего метода основывается на выводах, что использование дополнительной информация улучшает качество генерации данных. Графы это специфические

данные, которые обладают собственными свойствами, характеризующими внутреннее строение структуры, которые можно использовать при генерации. В качестве дополнительной информации будут использованы характеристики из теории графов, посчитанные на исходном графе. С этой точки зрения наш метод можно назвать само-условным GraphVAE.

2.2 Методы обработки графов

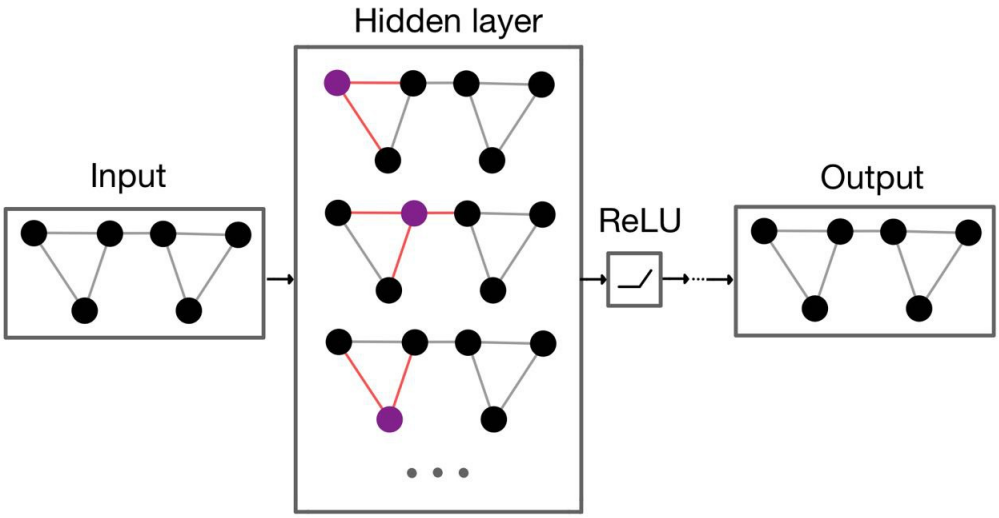

Рис. 3: Схема графовой сверточной нейронной сети

Одним из самых популярных методов обработки информации в графах является Graph Convolutional Networks (GCN). Он представляет из себя класс нейросетевых архитектур, которые применяются для анализа данных на графах. GCN является расширением сверточных нейронных сетей для данных, имеющих графовою структуру, которое учитывает матрицу смежности при свертке.

Метод был представлен в «Semi-Supervised Classification with Graph Convolutional Networks»[9]. Авторы отмечают, что прежде чем был предложен GCN для анализа графов применялись методы, такие как Graph Laplacian[10] и DeepWalk[11], которые имеют недостатки, связанные с ограничениями по сложности моделей и чрезмерно большим размером представлений графов.

Авторы приводят примеры использования GCN для классификации вершин в графах и для задачи сегментации изображений, а также выносят проблемы переобучение и предлагают методы регуляризации модели.

GCN объединяет свойства CNN и графовых моделей, которые позволяют применять сверточные операции непосредственно на графовых структурах. Важным понятием в GCN является операция Graph Convolution, которая агрегирует информацию соседних узлов графа и создает новое представление для текущего узла.

Формулы, описывающие работу GCN, выглядят следующим образом:

$$H^{[i+1]} = \sigma(W^{[i]} H^{[i]} A^*),$$

где $H^{[0]}$ – признаки вершин, A^* – нормализованная версия матрицы смежности A, $W^{[i]}$ – веса i-го слоя графовой свертки.

2.3 Генерация графов при помощи VAE

Генерация графов - это задача, которая привлекает внимание многих исследователей в области машинного обучения. В данном разделе разбирается один из самых популярных подходов к генерации графов, основанный на вариационном автокодировщике.

Variational autoencoders (VAE) - это класс моделей глубокого обучения, которые могут быть использованы для генерации графов. Основная идея заключается в том, чтобы отобразить данные в пространство меньшей размерности, в котором графы могут быть сгенерированы. Такое пространство называется латентным.

Одной из первых статей, посвященных генерации графов при помощи VAE, является «Variational Graph Auto-Encoders»[2]. Авторы предлагают использовать VAE для изучения скрытого распределения графов и генерации новых графов путем сэмплирования из этого распределения. В работе предложен графовый энкодер, который принимает матрицу смежности графа и преобразует ее в скрытый вектор. В декодере происходит обратное преобразование скрытого вектора в матрицу смежности.

Развитие этой идеи произошло в статье «GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders»[12], где VAE использовалась для генерации маленьких графов. Авторы предлагают модифицированный VAE, который использует два различных энкодера. Один для работы с вершинами графа, а другой - для работы с ребрами между вершинами. Также в статье предложена функция потерь, которая позволяет измерять качество генерации графов по нескольким метрикам.

Позже графовые VAE модели начали набирать популярность и использоваться во многих профильных областях. Так в статье «Junction Tree Variational Autoencoder

for Molecular Graph Generation»[13] использовалась VAE для генерации молекулярных графов. В ней предлагается способ использования деревьев связей для представления молекул. Также была разработана новая функция потерь, которая позволяет управлять генерацией молекулярных графов.

В последние годы было предложено множество модификаций VAE для решения этой задачи, а также различные функции потерь и метрики для оценки качества генерации графов.

Нами было принято решение использовать модель GraphVAE как базовый алгоритм для дальнейших улучшений. В отличии от CVAE, дополнительную информацию мы будем использовать только на этапе работы с латентными векторами.

3 Основные понятия

В генеративных моделях часто необходимо чтобы элементы в скрытом пространстве имели заданное распределение. Это может регулироваться метриками сходства распределения. Например, одной из таких метрик является KL-дивергенция или расстояние Кульбака-Лейблера, которая используется и в нашей работе.

Определение 3.1. *KL-дивергения – мера расхождения двух вероятностных распределений, характеризуемая следующей формулой:*

$$KL(P\|Q) = -\sum P(X)\log\frac{Q(x)}{P(x)}$$

3.1 Виды признаков и различия между ними

В нашей работы мы оперируем рядом введенных понятий, таких как «простые признаки», «сложные признаки» и «смешанные признаки». Данные понятия не являются общепринятыми.

Определение 3.2. *Простые признаки (или же простые статистики) в предложенном методе - это числовые характеристики используемые в теории графов, которые могут быть вычислены не более, чем за квадратичное время.*

В качестве простых признаков графа нами были выбраны следующие характеристики:

- Размерные показатели [$O(1)$]:
  - Число ребер
  - Число вершин
- Вершины специального вида [$O(V)$]:
  - Изолированные вершины – вершины без единого ребра
  - Висячие вершины – вершины с одним ребром
  - Промежуточные вершины – вершины с двумя ребрами
  - Вершины, связанные с каждой вершиной графа
- Статистики на степенях вершин [$O(V)$]:
  - Максимальная степень вершины
  - Средняя степень вершины
  - Медианная степень вершины
  - Модальная степень вершины
  - Стандартное отклонение степеней вершин в графе
- Гистограмма степеней вершин графа [$O(V)$] (здесь μ – средняя степень вершин в графе, σ – среднеквадратичное отклонение степеней вершин в графе): Доля вершин со степенью на интервалах: $(\mu - \sigma, \mu)$, $(\mu, \mu + \sigma)$, $(\mu - 2\sigma, \mu - \sigma)$, $(\mu + \sigma, \mu + 2\sigma)$, $(\mu - 3\sigma, \mu - 2\sigma)$, $(\mu + 2\sigma, \mu + 3\sigma)$
- Оценка размер наибольшей клики в графе [$O(Vd^2)$, d – максимальная степень вершины] [14]
- Коэффициент кластеризации [$O(V^2)$] [15]

Определение 3.3. *Смешанными признаками (или же смешанными статистиками) назовем любой способ численно описать граф.*

Замечание 3.1. В нашей работе мы будем рассматривать не все возможные численные представления графа, а только те, которые можно получить в модели VAE в скрытом представлении.

Под скрытым представлением обычно понимают сжатое представление входных данных, которого достаточно для их восстановления.

Определение 3.4. *Сложными признаками (или же сложными статистиками) назовем вектор $\vec{d}$, такой что каждая компонента вектора $\vec{d}$ статистически независима от компонент вектора простых статистик $\vec{s}$ и при этом вектор смешанных статистик m выражается через $\vec{d}$ и $\vec{s}$ линейно.*

Иными словами под сложными статистиками мы будем понимать те особенности графа, которые невозможно выразить при помощи простых статистик.

Замечание 3.2. В общем смысле все статистики графа представляют собой некоторые функции, которые переводят граф в действительное числовое пространство. В нашей работе мы будем подразумевать под той или иной статистикой реализацию функции на заданном графе.

3.2 Распределение графа

Поскольку мы будем работать не с детерменированными, а вероятностными моделями, важно определить, что будет подразумеваться под распределением некоторого графа G.

Определение 3.5. *Пусть граф G имеет матрицу смежности A. Распределением графа G будет называть $\{\hat{G}_i\}$ – множество графов (и соответствующие им матрицы смежности $\{\hat{A}_i\}$, полученных небольшими флуктуациями матрицы A.*

Флуктуацией графа будем называть изменение $A_{ij} = 0$ на $A_{ij} = 1$ и наоборот. Под степенью флуктуации будет пониматься пара из вероятностей, отвечающая за изменение в матрице смежности нуля на единицу и единицы на ноль, то есть за появление или исчезновение ребра в графе.

Замечание 3.3. Если отождествлять распределение графа с одномерной случайной величиной, то под распределением графа G можно понимать распределение нормированного отличия исходной матрицы смежности A от измененных матриц $\{\hat{A}_i\}$.

Кроме того, если у графа G есть некоторое числовое описание, то распределением такого описания, будет являться множество числовых описаний графов $\{\hat{G}_i\}$.

4 Постановка задачи

В ходе решения проблемы генерации, возникает две задачи – задача правильного восстановления матрицы смежности из скрытого пространства и задача построения неизвестного распределения данных.

4.1 Задача реконструкции

Традиционно, для того, чтобы обучить модель генерировать новый элемент данных, необходимо научиться реконструировать объект из скрытого пространства. Это так называемое end-to-end обучение. В нашем случае мы будет предсказывать наличие и отсутсвие ребра в графе.

Формально постановка этой задачи может быть описана следующим образом.

Дано:

Граф G с матрицей смежности $A \in \mathcal{R}^{n \times n}$, где $A_{ij} = 1$, если ребро (i, j) существует в графе, и 0 в противном случае. Матрица признаков вершин $V_f \in \mathcal{R}^{n \times k}$, а также набор скрытых ребер $E = \{(i, j)\}$. Здесь k – размер вектора признаков вершин.

Задача:

Построить модель, предсказывающую наличие ребра в графе на основе признаков вершин и существующих ребер. Однако особый интерес представляет предсказание наличия маркированных ребер. Задача может быть сформулирована как задача бинарной классификации: для каждой пары вершин i и j нужно предсказать вероятность того, что ребро (i, j) существует в графе, то есть принимает значение 1 в матрице смежности.

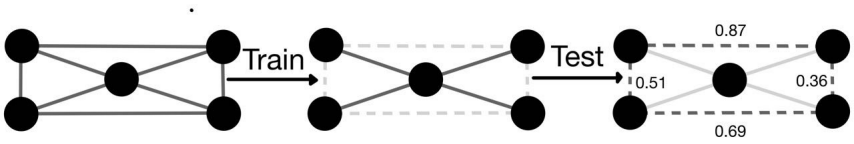

Рис. 4: Процесс маркировки ребер при генерации и тестировании модели

Модель обучается на данных, которые представляют собой множество пар вершин с маркированными и немаркированными ребрами. Она должна определить какие признаки графа могут помочь в предсказании наличия ребер. На основе этих признаков необходимо построить модель которая может классифицировать каждую пару вершин в графе.

Результатом работы модели является матрица предсказанных вероятностей существования ребер между всеми парами вершин в графе, включая немаркированные ребра.

4.2 Задача генерации

Задача по получению новых графов из неизвестного распределения выглядит следующим образом.

Дано:

Множество $\{A_i\}_{i=0}^{N} \in \mathcal{R}^{n \times n}$ матриц смежности графов G_i из неизвестного распределения $\pi(G)$, построенного на основе графа G.

Задача:

Получить распределение $\pi(G)$ в целях оценки $\pi(\hat{G})$ для нового графа $\hat{G}$ и генерации новых графов из распределения $\pi(G)$.

5 Предлагаемый метод

Наш метод генерации графов основан на модели графового вариационного автоэнкодера (GraphVAE). Основным преимуществом нового подхода является возможность контролировать генерацию, задавая определенные свойства графа. Это делает его более удобным и гибким по сравнению с обычным VAE.

5.1 Описание метода

Мы разделяем матрицу признаков графа на «простые статистики» и «сложные статистики». Простые статистики, которые, как сказано в формальном определении, представляют собой свойства графа, которые мы вычисляем заранее эффективными детерминированными алгоритмами. Все используемые простые статистики приведены в разделе 3.1. Сложные статистики, напротив, представляют собой набор характеристик, которые либо вычисляются алгоритмами с более высокой сложностью, либо не вычисляются при обычном анализе графа. К примеру, максимальная центральность смежности в графе и длина максимального цикла в графе или производные от них.

Поскольку мы работаем с вероятностной моделью, нам необходимо получить распределение графа. Поэтому, мы проводим следующую операцию с исходным гра-

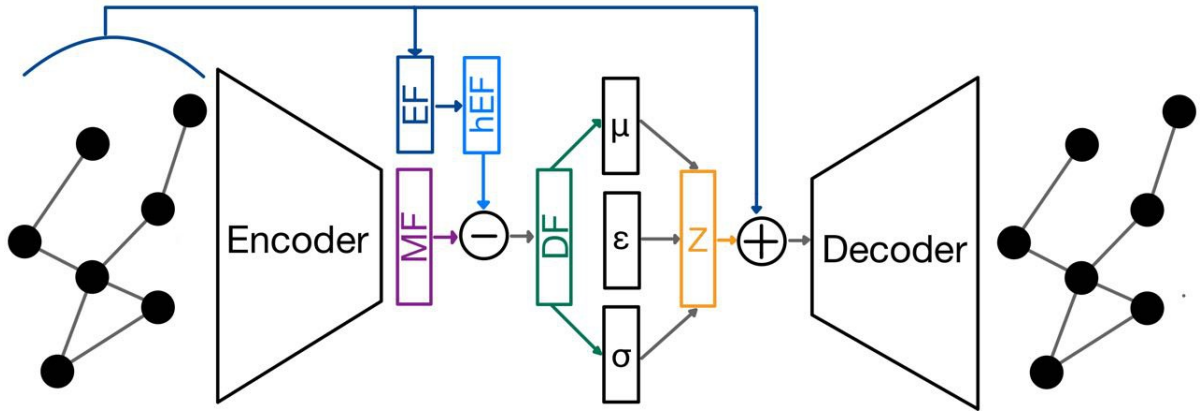

Рис. 5: Схема предложенного метода. Здесь MF – матрица смешанных статистик графа, EF – вектор простых статистик, hEF – скрытое представление вектора простых статистик, DF – матрица сложных статистик, ε – случайная величина $\in N(0,1)$, а Z – матрица из распределения $N(\mu,\sigma)$

фом G – мы либо добавляем, либо убираем часть ребер графа, тем самым получая граф, похожий на исходный, но все же имеющий изменения в матрице смежности. Такую операцию мы проводим несколько раз, тем самым получая набор графов. Для каждого нового графа мы рассчитываем простые статистики, а также переводим каждый из графов в скрытое представление путем использования графовых сверток.

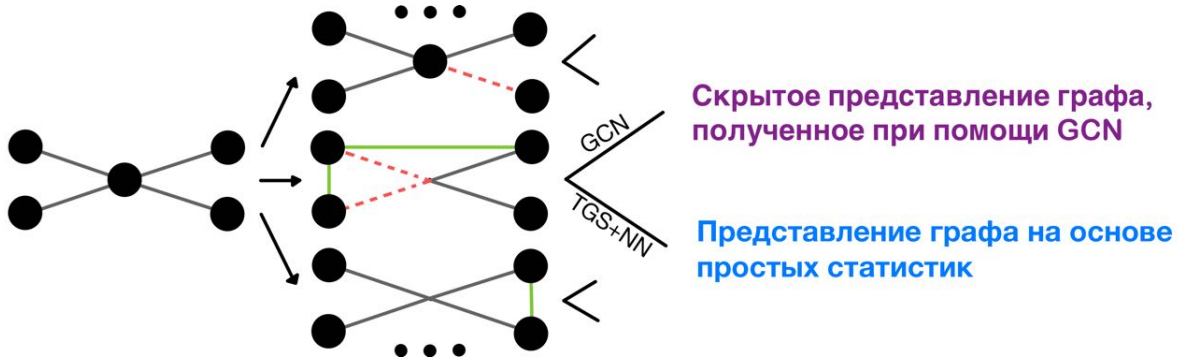

Рис. 6: Процесс создания распределения графа на основе имеющегося

Как говорилось ранее, мы отождествляем скрытое представление графа с смешанными статистиками. Также мы переводим простые статистики в некоторое скрытое пространство линейной сверткой, требуя, чтобы получившееся представление имело нормальное распределение с нулевым математическим ожиданием, а линейные комбинации его компонент наилучшим образом приближали смешанные статистики в скрытом представлении графа.

Таким образом, под распределением скрытых представлений (они же смешан-

ные статистики) и распределением простых статистик, будем понимать множество скрытых представлений и векторов простых статистик, соответствующих графам из распределения графа G.

Скрытое представление может характеризоваться несколькими векторами скрытого пространства. Их число является параметром модели и регулируется блоками с графовыми свертками. В тоже время вектор простых статистик является единственным для графа. Для того, чтобы учитывать влияние простых статистик на каждый из векторов скрытого пространства, мы размножим его дублированием на число, равное числу векторов в скрытом пространстве. Получившуюся таким образом матрицу будем называть матрицей простых статистик.

Следующим шагом алгоритма, мы вычитаем оценку смешанных статистик, полученную линейным преобразованием простых, из самих смешанных статистик. Тем самым мы оставляем в скрытом представлении только те характеристики графа, которые нельзя линейно выразить через простые.

И наконец, мы получаем математическое ожидание и дисперсию из матриц, полученных на прошлом этапе вычитанием матрицы простых статистик из матрицы смешанных, путем нахождения среднего и среднеквадратичного отклонения для всех измененных графов. Затем мы генерируем матрицу сложных статистик, пользуясь трюком репараметризации. Далее добавляем к матрице сложных статистик графа матрицу простых статистик. Тем самым имея итоговую матрицу признаков графа, с помощью которой декодер формируем матрицу смежности.

Таким образом, мы генерируем матрицу на основе сгенерированных сложных признаков графов и фиксированных простых статистик. То есть мы явно фиксируем какими свойствами должен обладать сгенерированный граф, путем добавления этих свойств в вектор простых статистик.

Наш метод позволяет генерировать графы с более точными фиксированными статистиками при большем разнообразии сгенерированных примеров. Это достигается за счет контроля за процессом генерации, а также за счет улучшенного скрытого пространства. Таким образом, наш метод имеет большой потенциал для использования в различных областях где требуется генерация графов с заданными свойствами.

5.2 Теоретическое обоснование

Вариационный автокодировщик (VAE) — это генеративная модель, которая обучается отображать объекты в заданное скрытое пространство, после чего генерировать новые объекты из этого скрытого пространства.

Часто важно, чтобы элементы скрытого пространства были распределены стандартно нормально. В нашей работе мы достигаем это стандартизацией скрытого представления по всем скрытым представлениям, полученным от преобразованных графов.

Далее в этом разделе под простыми статистиками будем понимать скрытое представление простых статистик, которое также будет иметь нормальное распределение с нулевым математическим ожиданием. Это будет достигаться теми же способами, которые используются для получения нормально распределенных смешанных статистик.

Наша цель – разложить смешанные статистики графа G в линейную комбинацию независимых друг от друга простых и сложных статистик.

Замечание 5.1. Далее будет рассматриваться лишь один из векторов смешанных статистик. Однако рассуждения можно провести для каждого вектора из матрицы смешанных статистик.

Пусть у нас есть вектор смешанных статистик $\vec{m}$ и вектор простых статистик $\vec{s}$, такие, что $|\vec{m}| > |\vec{s}|$, принадлежащие соответствующим распределениям. Причем оба вектора состоят из независимых одинаково (стандартно нормально) распределенных случайных величин.

Замечание 5.2. Статистическую независимость элементов вектора $\vec{s}$ мы можем гарантировать по построению.

В силу статистической независимости элементов векторов, ясно, что не существует линейного отображения из вектора $\vec{s}$ в вектор $\vec{m}$. Однако обратное утверждать нельзя, поэтому выдвинем следующую гипотезу.

Гипотеза 5.1. *Существует линейное отображение вектора смешанных статистик в вектор простых статистик.*

Иными словами, существует матрица $A^{|\vec{s}\times\vec{m}|}$, задающая следующее отображение: $A\vec{m} = \vec{s}$.

Введем следующую лемму:

Лемма 5.1. *Пусть дан набор независимых, одинаково распределенных (нормально) случайных величин* $p_1, p_2, \dots p_n$. *Случайная величина* $\xi = a_1p_1 + a_2p_2 + \dots + a_np_n$ *статистически зависима от каждой из случайных величин* p_i, *коэффициент перед которой* $a_i \neq 0$.

Доказательство.

Докажем это утверждение для p_1.

В силу теоремы[16] о сохранении нормальности при линейном преобразовании, случайная величина ξ будет также иметь нормальное распределение.

Для нормальных величин существует критерий независимости, который можно записать для p_i и ξ:

$$\mathbb{E}(p_i\xi) - \mathbb{E}(p_i)\mathbb{E}(\xi) = 0$$

Распишем этот критерий, воспользовавшись тем фактом, что p_i и p_j являются независимыми для $\forall i \neq j$:

$$a_1\mathbb{E}(p_1^2) + a_2\mathbb{E}(p_1)\mathbb{E}(p_2) + \dots + a_n\mathbb{E}(p_1)\mathbb{E}(p_n) - \mathbb{E}(p_1)(a_1\mathbb{E}(p_1) + \dots + a_n\mathbb{E}(p_n)) = 0$$

$$a_1\mathbb{D}(p_1) \neq 0$$

Таким образом в силу критерия независимости случайных величин, ξ статистически зависима от каждого слагаемого, коэффициент при котором не равен нулю. ■

Лемма 5.2. *Матрица A имеет максимально возможный ранг.*

Доказательство.

Докажем это от противного – в матрице A есть линейно зависимая строка.

Для простоты дальнейших выкладок, пусть есть строка, которая является линейной комбинацией двух других, которые между собой линейно не зависят. Каждой из этих строк соответствуют компоненты вектора $\vec{s}$:

$$\begin{cases} \xi = k_1\eta + k_2\zeta, \\ \eta = a_1p_1 + \dots a_np_n \\ \zeta = b_1p_1 + \dots n_np_n. \end{cases}$$

ξ, η, ζ будут нормальными случайными величинами в силу теоремы о сохранении нормальности при линейном преобразовании[16].

Воспользуемся критерием независимости нормальных случайных величин для ξ и η.

$$\mathbb{E}(\xi\eta) = \mathbb{E}((k_1\eta + k_2\zeta)\eta) = k_1\mathbb{E}(\eta^2) + k_2\mathbb{E}(\zeta\eta) = k_1\mathbb{E}(\eta^2) + k_2\sum_{i=1}^{n} b_i\eta p_i$$

$$\mathbb{E}(\xi)\mathbb{E}(\eta) = (k_1\mathbb{E}(\eta) + k_2\mathbb{E}(\zeta))\mathbb{E}(\eta) = (k_1\mathbb{E}(\eta) + k_2\sum_{i=1}^{n} b_i\mathbb{E}(p_i))\mathbb{E}(\eta)$$

$$\mathbb{E}(\xi\eta) - \mathbb{E}(\xi)\mathbb{E}(\eta) = k_1(\mathbb{E}(\eta^2) - (\mathbb{E}(\eta))^2) + k_2\sum_{i=1}^{n} b_i(\mathbb{E}(p_i\eta) - \mathbb{E}(p_i)\mathbb{E}(\eta))$$

В силу леммы 4.1: $b_i(\mathbb{E}(p_i\eta) - \mathbb{E}(p_i)\mathbb{E}(\eta)) = b_ia_i\mathbb{D}p_i$. Тогда:

$$\mathbb{E}(\xi\eta) - \mathbb{E}(\xi)\mathbb{E}(\eta) = k_1\mathbb{D}(\eta) + k_2\sum_{i=1}^{n} a_ib_i\mathbb{D}p_i$$

Поскольку p_i распределены стандартно нормально:

$$\mathbb{E}(\xi\eta) - \mathbb{E}(\xi)\mathbb{E}(\eta) = k_1\mathbb{D}(\eta) + k_2\sum_{i=1}^{n} a_ib_i$$

В силу линейной независимости строк матрицы A, соответствующих η и ζ, $\sum_{i=1}^{n} a_ib_i = 0$. Тогда критерий преобразуется в вид:

$$\mathbb{E}(\xi\eta) - \mathbb{E}(\xi)\mathbb{E}(\eta) = k_1\mathbb{D}(\eta) \neq 0, \text{ поскольку } k_1 \neq 0 \text{ и } \mathbb{D}(\eta) > 0$$

Таким образом, ξ и η зависимые случайные величины, что противоречит условию о статистической независимости компонент вектора $\vec{s}$

■

На основе данного отображения построим следующие преобразования:

Дополним матрицу $A^{|\vec{s}\times\vec{m}|}$ до матрицы $\hat{A}^{|\vec{m}\times\vec{m}|}$. Существует бесконечно много способов дополнить матрицу таким образом. Для того, чтобы уйти от неоднозначности нужно наложить дополнительные ограничения, о которых мы скажем ниже.

После перехода к преобразованию, добавленную часть матрицы $\hat{A}$ будем называть матрицей $D_{\hat{A}}$, а изначальную $A_{\hat{A}}$. Кроме того, часть нового вектора $\vec{\hat{s}}$ будем называть $\vec{d}$.

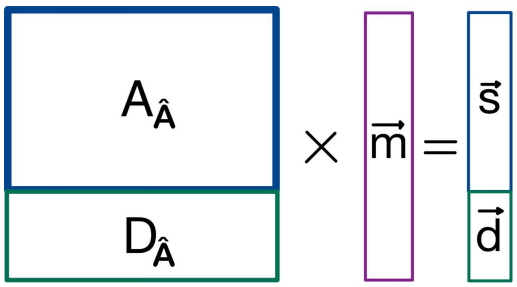

Рис. 7: Построенное линейное преобразование смешанных статистик

В качестве дополнительных ограничений на матрицу $D_{\hat{A}}$ примем требование на максимально возможный ранг матрицы $D_{\hat{A}}$ и линейную независимость каждой строки матрицы $D_{\hat{A}}$ от строк матрицы $A_{\hat{A}}$. В силу Леммы 4.2, матрица $D_{\hat{A}}$ будет ортогональным дополнением матрицы $A_{\hat{A}}$.

При линейной независимости строк матрицы $\hat{A}$, компоненты векторов $\vec{s}$ и $\vec{d}$ будут статистически независимы.

Таким образом, получившийся вектор $\vec{d}$ будет нормальным вектором в силу теоремы о сохранении нормальности при линейном преобразовании[16]. Также, каждая его компонента статистически независима от компонент вектора $\vec{s}$, а матрица $\hat{A}$ имеет полный ранг(а потому существует обратное преобразование). Итого получаем, что вектор $\vec{d}$ есть вектор сложных статистик по определению.

Теорема 5.1 (Бишук 2023). *Преобразование, описанное выше соответствует разложению смешанных статистик на простые и сложные.*

5.3 Использование в смежных областях

Разделение графов на «простые» и «сложные» признаки может быть полезным в различных областях, например:

- В *биоинформатике* графы могут представлять собой молекулы лекарственных препаратов. Простые признаки могут быть связаны с физико-химическими

свойствами атомов, а сложные с их биологическими свойствами, такими как взаимодействие с рецепторами.

- Графы *социальных сетей* могут быть использованы для анализа социальных взаимодействий и связей между людьми. Простые признаки могут быть связаны с такими характеристиками, как возрастные и половые группы, общительность, а сложные признаки - с социальным статусом, интересами и т.д.

- В *финансовой сфере* графы могут представлять собой финансовые потоки между компаниями. Простые признаки могут быть связаны с финансовыми показателями компаний, такими как источники доходов и расходов, а сложные признаки - с отношениями между компаниями, такими как поставщик-потребитель, конкуренты и т.д.

В качестве простых признаках предполагается использовать те свойства графа, которые важны в конкретной прикладной области (например, для графа контактов задать максимальную степень вершины), а остальные же взаимосвязи называть сложными. Такие сложные зависимости будут отличаться не только от области к области, но и от специфики конкретного набора данных. Например социальный граф конкретной сети.

6 Вычислительный эксперимент

В этом разделе мы производим проверку предложенного метода и сравниваем его с существующими методами на реальных данных.

6.1 Наборы данных

Эксперимент проводился на наборах данных «Cora» и «Citeseer». Представляющие собой информацию о научных статьях и их цитировании.

6.1.1 Cora

Датасет «Cora»[17] - это один из наиболее часто используемых датасетов в задачах классификации и кластеризации графов. Он состоит из 2708 статей, взятых из базы arXiv, и разбитых на 7 категорий: биология, информатика, право, математика,

медицина, физика и социология. Каждая статья представлена в виде узла графа, а связи между статьями - это ссылки между ними.

Каждая статья представлена в виде признаков, являющимися мешком слов (bag-of-words) из 1433 уникальных терминов, которые были извлечены из полного текста статей.

Таким образом датасет представляет из себя ориентированный граф, каждая вершина которого представлена набором из 1433 признаков.

6.1.2 Citeseer

В качестве второго датасета для тестирования нашего метода был выбран датасет «Citeseer»[18], который традиционно используется для задачи классификации статей по научным темам. Датасет содержит статьи из компьютерных наук и связанных с ними областей, таких как базы данных, информационный поиск и машинное обучение. Каждая статья представляет собой узел в графе, а ссылки на другие статьи формируют ребра. В датасете всего 6 классов научных тем: базы данных, интеллектуальная обработка информации, машинное обучение, информационный поиск, распределенные системы и робототехника. Он содержит 3327 статьи и 9228 ссылки между ними. Каждая статья представлена в виде метаданных, включающих название, список авторов, перечень ссылок на другие статьи и аннотацию.

Данные датасета были собраны из различных источников, включая базы данных ACM, DBLP и PubMed. В оригинальной статье, описывающей датасет, авторы провели анализ структуры сети цитирования, выделили основные сообщества статей и оценили качество работы алгоритмов кластеризации и классификации на этих данных.

6.2 Процесс обучения и тестирования

Обучение происходило на популярных датасетах Cora, Citeseer. Для обучения использовалась матрица смежности графов датасета, а также матрица признаков вершин.

Для того, чтобы не возникало взрыва и затухания градиента необходимо нормализовать матрицу смежности. Эффективность этого была доказана в работе, посвященной графовому автокодировщику[9]. Причем авторами утверждается, что наиболее эффективной является нормировка следующего вида:

$$\hat{A} = D^{-1/2} A D^{-1/2}, \text{ где}$$

- A – матрица смежности графа,
- D – диагональная матрица степеней вершин.

Наша модель получает на вход матрицу смежности графа, признаки вершин, а также вектор простых статистик, которые вычислялись детерминированными алгоритмами.

В энкодере, после применения блоков GCN из получившегося представления графа вычитается, преобразованный в скрытое пространство размноженный вектор простых статистик. Особенность скрытого пространства простых статистик заключается в том, что оно должно быть как можно ближе к скрытому представлению графа. Для этого мы рассчитываем среднеквадратичное отклонение между смешанным представлением графа и матрицей простых статистик, а при обратном проходе градиента, стремимся его уменьшить.

С каждой итерацией обучения алгоритма мы приближаем скрытый вектор простых статистик к вектору смешанных статистик. Таким образом, в скрытом векторе графа останется только та информация, которую нельзя получить при помощи простых статистик, то есть вектор сложных статистик.

Теперь чтобы получить скрытый вектор для генерации нового графа, мы пропускаем вектор сложных статистик через блок GCN и получаем параметры для нормального распределения, которое соответствует сложных статистикам поданного на вход графа. После генерации скрытого вектора статистик из нормального распределения, в котором теперь должны находиться только сложные статистики, мы суммируем его с преобразованными в скрытое представление простыми статистиками. На основе полученной матрицы, мы создаем матрицу смежности. В ячейке этой матрицы находится вероятность того, что на этом месте должно быть ребро.

На каждой эпохе часть ребер случайным образом выбиралось и скрывалось. Кроме того выбиралось такое же количество случаев, когда между вершинами ребра не существовало. После этого эти ребра разделяются на группы для обучения, валидации и теста.

После прохода нейронной сети и получения матрицы смежности, рассчитывался BCELoss на скрытых ребрах, а также на всей матрицы смежности в целом. Также

прибавлялась KL-дивергенция между распределением скрытого вектора сложных статистик и стандартным нормальных распределением и MSELoss с этапа приближения простых признаков.

Итоговая функция потерь для нашей модели выглядела следующим образом

$$\text{Loss}_{method}(Y, \hat{Y}, M, S, D) = -\frac{1}{N^2}\sum_{i=0}^{N}\sum_{j=0}^{N} w_{ij}[Y_{ij}\log(\hat{Y}_{ij}) + (1 - Y_{ij})\log(1 - \hat{Y}_{ij})]$$

$$-KL\_div(D, N(0,1)) - \frac{1}{N}\sum_{i=0}^{N}\sum_{j}(M_{ij} - S_j)^2$$

где:

- Y – матрица смежности входного графа;
- $\hat{Y}$ – матрица смежности сгенерированного графа, где в ячейке $\hat{Y}_{ij}$ стоит вероятность того, что существует ребро между вершинами i и j;
- M – матрица смешанных статистик,
- S – вектор простых статистик входного графа
- D – матрица сгенерированных сложных статистик
- w_{ij} – веса перед элементами в матрице смежности. Самый большой вес у маркированных ребрах, затем у маркированных мест, где ребра нет, меньший вес у существующих не маркированных ребер и самый маленький вес у мест, где ребер нет и они не маркированы. Вес зависит от того было ли место в матрице маркировано и сколько всего ребер есть в рассматриваемом графе.

6.3 Результаты

Поскольку результатом работы алгоритма является матрица с вероятностями, традиционно для классификации рассматривают метрики качества ROC-AUC и Average Precision, поскольку они не требуют выбора порога бинаризации и более комплексно оценивают качество классификации.

Однако для подсчета разницы между статистиками поданного на вход и сгенерированного графов, нам необходимо бинаризовать сгенерированную матрицу. Для

этого мы проходим со всеми возможными порогами для вероятности в матрице смежности сгенерированного графа и рассчитываем простые статистики. После чего мы выбираем тот порог, который соответствует минимуму средней абсолютной ошибки между простыми статистиками входного графа и сгенерированного.

Такой минимум мы считаем лучшим результатом, который может получить алгоритм генерации и записываем его в таблицу результатов 6.3.

	Dataset	ROC-AUC	AP	MAE (global statistics)
GraphVAE	Cora	75.18 %	**75.81%**	0.066
Our method		**76.68 %**	75.18 %	**0.046 (-30%)**
GraphVAE	Citeseer	**82.09 %**	**79.94 %**	0.072
Our method		76.28 %	76.44 %	**0.060 (-17%)**

Таблица 1: Результаты вычислительного эксперимента по классификации наличия ребер.

Как видно из таблицы 6.3, предложенный метод генерирует графы с статистиками, более близкими к исходным, чем обычная модель GraphVAE.

Мы полагаем, что уменьшение метрик реконструкции можно нивелировать, используя другой подход к агрегации простых признаков, либо более тонкой настройкой параметров обучения. Подробнее этот вопрос будет исследован в будущих работах.

7 Заключение

В ходе данной работы был разработан и теоретически обоснован новый метод генерации графов, использующий идею разделения статистик графа на простые (легко вычислимые и интерпретируемые) и сложные. Эксперименты, проведенные на популярных датасетах Cora и Citesser, показали эффективность предложенного метода в сравнении с оригинальной моделью графового VAE.

Небольшое снижение метрик реконструкции оставляет модель на уровне актуальных моделей оригинальной архитектуры, но при этом позволяет генерировать графы с заранее заданными свойствами.

Предложенный подход можно обобщить на любые модели, преобразующие данные в некоторое скрытое представление. Это может быть полезным, например, при

генерации молекулярных структур или сетей связей между людьми.

В будущих работах мы планируем расширить предложенный метод используя дополнительно новые простые статистики; исследовать влияние различных простые статистик на ограничение свободы генерации графов; рассмотреть разнообразные методы агрегации графов в векторе одной вершины.

Список литературы

[1] Zhitao Ying и др. «Gnnexplainer: Generating explanations for graph neural networks». В: *Advances in neural information processing systems* 32 (2019).

[2] Thomas N Kipf и Max Welling. «Variational graph auto-encoders». В: *arXiv preprint arXiv:1611.07308* (2016).

[3] Ben Chamberlain и др. «Grand: Graph neural diffusion». В: *International Conference on Machine Learning*. PMLR. 2021, с. 1407—1418.

[4] Vito Latora и Massimo Marchiori. «A measure of centrality based on network efficiency». В: *New Journal of Physics* 9.6 (2007), с. 188.

[5] Diederik P Kingma и Max Welling. «Auto-encoding variational bayes». В: *arXiv preprint arXiv:1312.6114* (2013).

[6] Kihyuk Sohn, Honglak Lee и Xinchen Yan. «Learning structured output representation using deep conditional generative models». В: *Advances in neural information processing systems* 28 (2015).

[7] Lecun Yann. «The mnist database of handwritten digits». В: *R* (1998).

[8] Alex Krizhevsky, Geoffrey Hinton и др. «Learning multiple layers of features from tiny images». В: (2009).

[9] Thomas N Kipf и Max Welling. «Semi-supervised classification with graph convolutional networks». В: *arXiv preprint arXiv:1609.02907* (2016).

[10] Russell Merris. «Laplacian matrices of graphs: a survey». В: *Linear algebra and its applications* 197 (1994), с. 143—176.

[11] Bryan Perozzi, Rami Al-Rfou и Steven Skiena. «Deepwalk: Online learning of social representations». В: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, с. 701—710.

[12] Martin Simonovsky и Nikos Komodakis. «Graphvae: Towards generation of small graphs using variational autoencoders». В: *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part I 27*. Springer. 2018, с. 412—422.

[13] Wengong Jin, Regina Barzilay и Tommi Jaakkola. «Junction tree variational autoencoder for molecular graph generation». В: *International conference on machine learning*. PMLR. 2018, с. 2323—2332.

[14] Bharath Pattabiraman и др. «Fast algorithms for the maximum clique problem on massive sparse graphs». В: *Algorithms and Models for the Web Graph: 10th International Workshop, WAW 2013, Cambridge, MA, USA, December 14-15, 2013, Proceedings 10*. Springer. 2013, с. 156—169.

[15] Jari Saramäki и др. «Generalizations of the clustering coefficient to weighted complex networks». В: *Physical Review E* 75.2 (2007), с. 027105.

[16] Александр Алексеевич Боровков. *Теория вероятностей*. URSS, 2009.

[17] Prithviraj Sen и др. «Collective classification in network data». В: *AI magazine* 29.3 (2008), с. 93—93.

[18] Ryan Rossi и Nesreen Ahmed. «The Network Data Repository with Interactive Graph Analytics and Visualization». В: *AAAI Conference on Artificial Intelligence*. Т. 29. New York, NY, USA, 2015, с. 4292—4293.