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

Контролируемая генерация графов

03.03.01 — 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 статистических характеристик генерируется в скрытом пространстве и затем используется для формирования матрицы смежности графа. 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 сходную структуру с исходным графом, что особенно полезно при работе с графами, описывающими контакты между людьми, например, граф контактов сотрудников офиса.

1 Введение

Все более популярным становится использование графов в качестве источников данных в задачах машинного обучения, однако остается недостаточно развито в силу отсутствия достаточно больших наборов данных для обучения и тестирования моделей. Для решения этой проблемы часто используются задачи, основанные на единственном графе, либо на синтетических генераторах графов [1]. Однако эти методы могут быть не неточными или не всегда коррелирующими с реальными данными.

Кроме того, часто возникает потребность в графах, имеющих схожее распределение с имеющимся. Это важно, например, в случае графа контактов фиксированного сообщества, где необходимо сгенерировать ряд похожих на исходных графов. Для этой задачи традиционно используются генеративные модели, однако существующие методы могут ограничивать возможности генерации и не учитывать интуитивно понятные характеристики, такие как число клик или число рёбер.

Для такого рода задач традиционно используют генеративные модели. Для этого подходят как простые модели, такие как GraphVAE [2], так и диффузионные [3].

Однако существующие методы так или иначе сильно ограничивают возможности генерации. Они фокусируются на реконструкции исходного графа, и не способны учитывать при генерации интуитивно понятные характеристики (в случае графа контактов, например, время общения человека в офисе ограничено, а потому есть ограничения на время и число контактов). Мы же предлагаем использовать в качестве ограничения на генерацию не сколько качество реконструкции, сколько заранее выбранные глобальные статистики графа (например число ребер, вершин, кластерное число и так далее). Тем самым мы можем генерировать графы с заранее выбранными статистиками, но имеющими схожее распределение с исходным графом.

Кроме того, этот метод может быть использован для поиска «сложных» статистик графа, таких как центральность смежности [4], которые до сих пор остаются нерешенными на достаточно высоком уровне. Подробнее об использовании подхода в других задачах мы расскажем в разделе с методом.

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

2 Обзор литературы

Генерация графов - это активно развивающийся направление в области машинного обучения, которое находит применение во многих областях, например, биоинформатику, физику, NLP и многие другие. Генерация графов может помочь решать различные задачи, такие как поиск наиболее важных узлов в графе, классификация графов, прогнозирование свойств графов и многие другие. Существует множество методов 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], представляющей собой первое упоминание модели вариационных автокодировщиков (VAE).

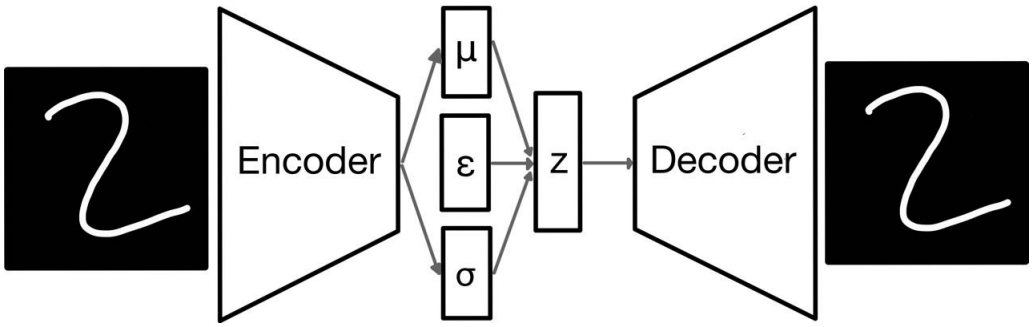

Рис. 1: Схема модели вариационного автокодировщика(VAE). Энкодер задает распределение $p_\varphi(Z|X)$, а декодер $q_\theta(\hat{X}|Z)$, где X и $\hat{X}$ реальный и сгенерированный объекты соответственно; $\varepsilon \in N(0,1)$.

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 представляют новый подход уменьшения 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 и открыла новые возможности для генеративного моделирования данных, включая генерацию графов при помощи вариационных автокодировщиков.

Наш метод основывается на использование дополнительной информации при генерации. Первое упоминание такой идеи было в статье «Learning Structured Output Representation using Deep Conditional Generative Models» [**sohn2015learning**], где, впервые была представлена модель Conditional Variational Autoencoders (CVAE).

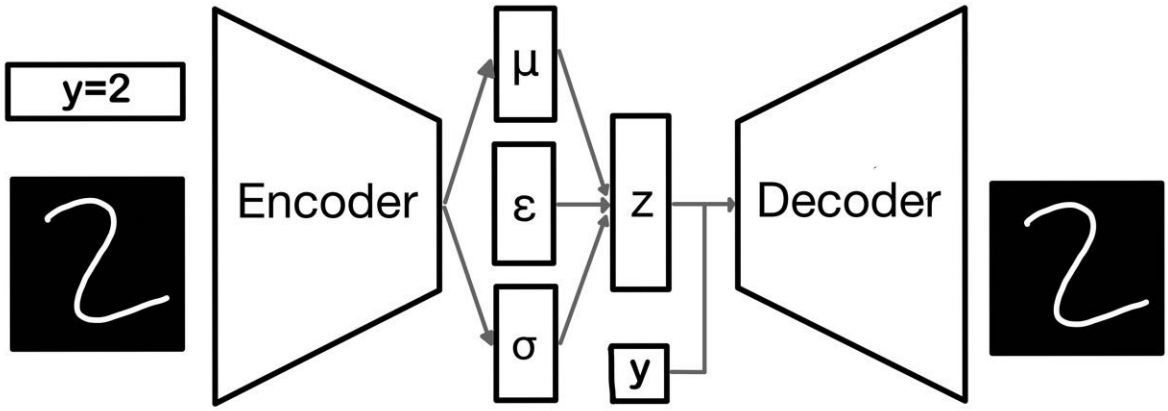

Рис. 2: Схема модели условного вариационного автокодировщика (CVAE). Энкодер задает распределение $p_\varphi(Z|X,y)$, а декодер $q_\theta(\hat{X}|Z,y)$, где X и $\hat{X}$ реальный и сгенерированный объекты соответственно; y – метка объекта; $\varepsilon \in N(0,1)$.

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 использован для генерации изображений с заданными свойствами. Они используют MNIST [6] для генерации цифр с определенными свойствами, такими как цвет и положение цифры на изображении. Кроме того описывается, как CVAE может быть использован для классификации изображений. Авторы статьи применяют CVAE к задаче классификации CIFAR-10 [7], показывая, что CVAE может значительно улучшить точность классификации. Впоследствии было предложено множество модификаций идей CVAE, таких как AC-GAN (Auxiliary Classifier GAN) и InfoGAN (Information Maximizing GAN), которые используют по-

добные идеи для генерации изображений с более сложными свойствами.

Идея нашего метода основывалась на выводах, что подмешенная дополнительная информация помогает при генерации данных. Однако графы это специфические данные, которые обладают собственными свойствами, характеризующими внутреннее строение графа, которые можно использовать при генерации. В графовыми данными нет нужды использовать дополнительную информацию извне, поскольку мы можем выделить ее из наших же данных. С этой точки зрения наш метод можно назвать Self-Conditional GraphVAE.

2.2 Методы обработки графов

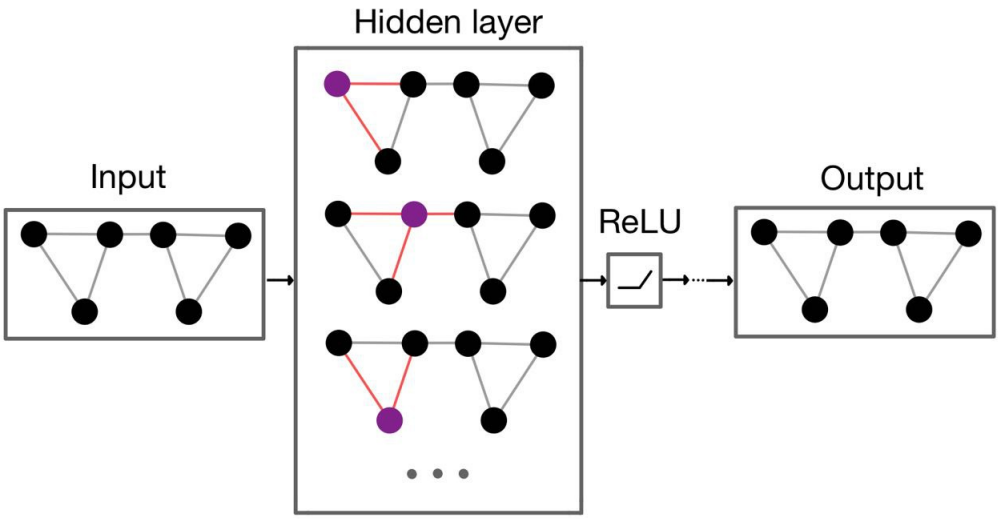

Рис. 3: Схема графовой сверточной нейронной сети

Одним из самым популярным методом 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, путем учета матрицы смежности.

Метод был представлен в «Semi-Supervised Classification with Graph Convolutional Networks» [8]. Авторы отмечают, что прежде чем GCN были предложены, для анализа графов применялись методы, такие как Graph Laplacian [9] и DeepWalk [10], которые имеют недостатки, связанные с ограничениями по сложности моделей и чрезмерно большим размером представлений графов.

Авторы привели примеры использования GCN для классификации вершин в графах и для задачи сегментации изображений, а также обсудили проблемы переобучение и предложили 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.

В целом, GCN является мощным инструментом для анализа графов и может быть применен в различных областях, включая социальные сети, биоинформатику, рекомендательные системы и многие другие.

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 последние годы было предложено множество методов для генерации графов, используя различные подходы.

Variational autoencoders (VAE) - это класс моделей глубокого обучения, которые могут быть использованы для генерации графов. Основная идея заключается в том, чтобы скрыть некоторую структуру, называемую латентным пространством, в котором графы могут быть сгенерированы.

Одной из первых статей, посвященных генерации графов при помощи VAE является «Variational Graph Auto-Encoders» [2]. Авторы предложили использовать VAE для изучения скрытого распределения графов и генерации новых графов путем сэмплирования из этого распределения. В работе был 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» [11], где VAE использовалась для генерации

маленьких графов. Авторы предложили модифицированный VAE, который использует два различных энкодера, один для работы с вершинами графа, а другой - для работы с ребрами между вершинами. Также в статье была предложена функция потерь, которая позволяет измерять качество генерации графов по нескольким метрикам.

Далее графовые VAE модели начали набирать популярность и использоваться во многих профильных областях. Так в статье «Junction Tree Variational Autoencoder for Molecular Graph Generation» [12] использовалась VAE для генерации молекулярных графов. В статье был предложен способ использования деревьев соединений для представления молекул, и была разработана новая функция потерь, которая позволяет управлять генерацией молекулярных графов.

В целом, использование VAE для генерации графов является перспективной идеей, и в последние годы было предложено множество модификаций VAE для решения этой задачи. Также были предложены различные функции потерь и метрики для оценки качества генерации графов.

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

3 Основные понятия

В этом разделе мы введем понятия, которые далее будем использовать.

3.1 Используемые понятия

В нашей работы мы оперируем рядом введенных понятий, таких как «простые признаки», «сложные признаки» и «смешанные признаки». Их мы определяем следующим образом:

Определение 3.1. *Простые признаки (или же простые статистики) в нашем методе - это признаки графа, которые могут быть вычислены не более, чем за квадратичное время.*

Уточняя данное определение, стоит сказать, что нам интересны величины, которые принято использовать в теории графов в качестве численного описания графа.

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] [13]
- Коэффициент кластеризации $[O(V^2)]$ [14]

Определение 3.2. *Смешанными признаками 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.

Определение 3.3. *Сложными статистиками назовем такие функции от графа, которые:*

1. *Линейно выразимы через смешанные статистики;*
2. *Линейно невыразимы через простые статистики;*
3. *Статистически независимы от простых статистик.*

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 некоторое действительное числовое пространство. В нашей работе мы будем подразумевать по той или иной статистикой реализацию функции на заданном графе.

Кроме того, в генеративных моделях часто необходимо, чтобы элементы в скрытом пространстве имели заданное распределение. Часто это регулируется метриками сходства распределения. Например одной из таких метрик является KL-дивергенция или расстояние Кульбака-Лейблера, которой пользуемся и мы в нашей работе.

Определение 3.4. *KL-дивергения – мера расхождения двух вероятностных распределений, характеризуемая следующей формулой:*

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

4 Предлагаемый метод

Наш метод генерации графов, основанный на модели вариационного автоэнкодера (VAE). Основное преимущество нового метода заключается в том, что он позволяет контролировать генерацию графов задавая ряд определенных свойств графа. Это делает его более удобным и гибким по сравнению с обычным VAE.

4.1 Описание метода

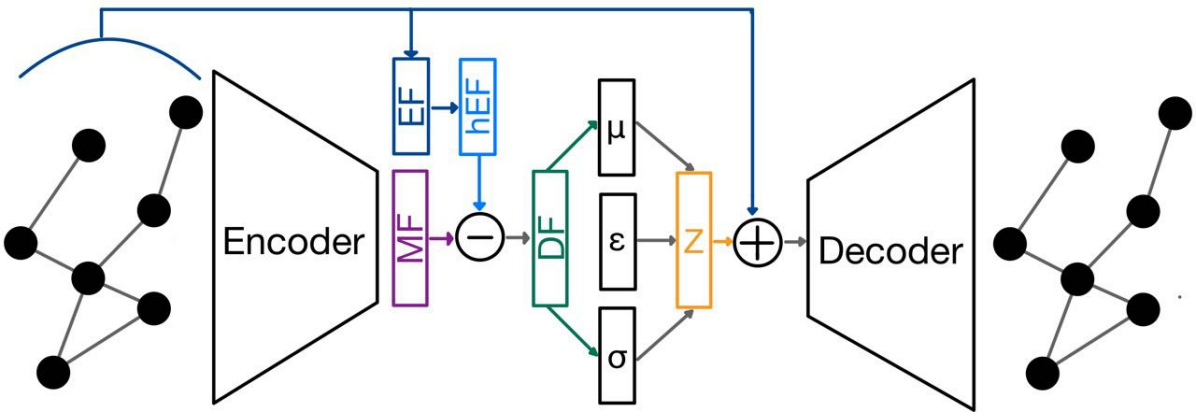

Рис. 4: Схема предложенного метода. Здесь MF – матрица смешанных статистик графа, EF – вектор простых статистик, hEF – скрытое представление вектора простых статистик, DF – матрица сложных статистик, ε – случайная величина $\in N(0,1)$, а Z – матрица из распределения $N(\mu, \sigma)$

В нашем методе мы разделяем матрицу признаков графа на «простые статистики» и «сложные статистики». Простые статистики, которые, как сказано в формально 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 вовсе. К примеру, максимальная центральность смежности в графе, длина максимального цикла в графе, либо производные от них.

Для этого мы проводим следующую операцию – мы маркируем часть ребер графа, тем самым получая граф, похожий на исходный, но все же имеющий небольшие изменения. Такую операцию мы проводим несколько раз, тем самым получая набор графов и для каждого нового графа мы рассчитываем простые статистики. После чего мы переводим каждый из графов в скрытое представление путем использования графовых сверток. Как говорилось ранее, мы отождествляем скрытое представление графа с смешанными статистиками. Аналогично мы переводим простые статистики в некоторое скрытое представление линейной сверткой, требуя, чтобы получившееся представление имело нормальное распределение с нулевым математическим ожиданием, а также наилучшим образом приближало смешанные статистики в скрытом представлении. После чего мы вычитаем оценку смешанных статистик, полученную

линейным преобразованием простых. Тем самым мы оставляем в матрице смешанных статистик только те характеристики графа, которые нельзя линейно выразить через простые.

Поскольку в скрытом представлении мы имеем несколько векторов, характеризующих граф, а вектор простых статистик один, мы размножим его дублированием. Это нужно, чтобы учесть одни и те же простых статистики для каждого вектора смешанных статистик. Получившуюся матрицу будем называть матрицей простых статистик.

Затем мы получаем математическое ожидание и дисперсию из матрицы, полученной на прошлом этапе путем нахождения среднего и среднеквадратичного отклонения для всем измененных графов. После чего генерируем матрицу сложных статистик, пользуясь трюком репараметризации. И наконец добавляем к матрице сложных статистик графа скрытую матрицу простых статистик. Тем самым мы получаем 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 большом 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.

4.2 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 нормально с нулевым математическим ожиданием. В нашей работе мы достигаем это стандартизацией скрытого представления по всем скрытым представлениям, полученным от преобразованных графов.

Кроме того 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. Причем далее будет рассматриваться лишь один из векторов смешанных статистик. Однако дальнейшие 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 (нормально) распределенных случайных величин с математическим ожиданием, равным нулю.

Замечание 4.1. Независимость элементов вектора s мы можем гарантировать по построению.

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

Гипотеза 4.1. *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 4.1. *Пусть дан набор независимых, одинаково распределенных 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 $a_i \neq 0$.*

Лемма 4.2. *Матрица A имеет полный ранг.*

Доказательство.

Предположим, что это нет так, тогда получается, что одна из строк матрицы A линейно зависима от других. Тогда это значит, что элемент в векторе $\vec{s}$, соответ-

ствующий этой строке также будет линейно зависим от других компонент. В силу Леммы 4.1 линейная зависимость влечет за собой статистическую зависимость, но по построению все компоненты вектора $\vec{s}$ независимы. Получили противоречие, а значит в матрице A нет зависимых строк, а значит она имеет полный ранг. ■

На основе данного отображения построим следующие преобразования:

Дополним матрицу $A^{|\vec{s}\times\vec{m}|}$ до матрицы $\hat{A}^{|\vec{m}\times\vec{m}|}$ так, чтобы существовало обратное преобразование. Существует бесконечно много способов дополнить матрицу таким образом. Для того, чтобы уйти от неоднозначности нужно наложить дополнительные ограничения.

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

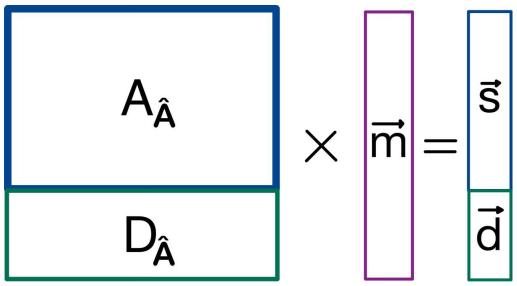

Рис. 5: Построенное линейное преобразование смешанных статистик

Каждый элемент вектора смешанных и простых статистик – это некоторая функция над графом G. Тогда запишем эти вектора в следующем виде:

$$m(\vec{G}) = (g_1(G), g_2(G), \ldots, g_{|m|}(G))$$

$$s(\vec{G}) = (f_1(G), f_2(G), \ldots, f_{|s|}(G))$$

Рассмотрим линейную оболочку над функциями смешанных статистик. Это линейное пространство размера $|m|$. В силу гипотезы существования отображения смешанных статистик в простые, функции $f_1(G), \ldots f_{|s|}(G)$ принадлежат этому линейному пространству.

Возьмем одну строку матрицы $A_{\hat{A}}$ и одну строку матрицы $D_{\hat{A}}$. Например первые строки каждой из матриц. Пусть они имеют вид $\hat{A}_1 = (\alpha_1, \alpha_2, \cdots, \alpha_{|m|})$ и $\hat{A}_{|s|+1} = (\beta_1, \beta_2, \cdots, \beta_{|m|})$, а соответствуют им значениям $s_1 = f_1(G)$, d_1. Иными словами $\hat{A}_1$ есть координаты функции s_1 в базисе функций смешанных статистик, аналогично

как и $\hat{A}_{|s|+1}$ есть координаты функции d_1.

Кроме того наложим на вектор $\vec{d}$ условие, что он независим от каждого элемента вектора $\vec{s}$. Иными словами необходимо, чтобы:

$$cov(s_i, d_j) = 0, \quad \forall i, j$$

Также это ограничение гарантирует также, что матрица $D_{\hat{A}}$ будет иметь полный ранг согласно Лемме 4.1.

Продолжим рассматривать элементы s_1 и d_1 подвекторов $\vec{s}$ и $\vec{d}$.

Поскольку вектор смешанных статистик $\vec{m}$ является стандартной нормальной случайной величиной, то в силу теоремы о линейном отображении нормальной случайной величины [15], получившийся вектор $\vec{\hat{s}}$ будет также иметь нормальное распределение с нулевым математически ожиданием. Кроме того, каждый подвектор вектора $\vec{\hat{s}}$ также будет иметь нормальное распределение с нулевым математическим ожиданием [15] – в том числе $\vec{s}$ и $\vec{d}$.

Вспомним также что, независимость двух случайных векторов с нулевым математическим ожиданием эквивалентно нулевой корреляции между векторами [15]. Тогда условие независимости записывается следующим образом:

$$cov(s_1, d_1) = \rho(s_1, d_1) = \alpha_1\beta_1 + \alpha_2\beta_2 + \ldots + \alpha_{|m|}\beta_{|m|} = 0$$

Записанное выше выражение является ни чем иным, как скалярным произведением двух строк матрицы $\hat{A}$.

Таким образом, для того, чтобы $\vec{d}$ был вектором сложных статистик, необходимо, чтобы все строки матрицы $D_{\hat{A}}$ были ортогональны всем строкам матрицы $A_{\hat{A}}$, а также $D_{\hat{A}}$ имела полный ранг. Поскольку в этом случае, элементы вектора d линейно выразимы из вектора смешанных статистик, линейно невыразимы из вектора простых статистик, а также от простых статистик не зависят, что в точности совпадает с определением сложных статистик.

Теорема 4.1 (Бишук 2023). *Преобразование, описанное выше соответствует разложению смешанных статистик на простые и сложные.*

4.3 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, где простые признаки могут быть связаны с физико-химическими свойствами атомов, а сложные признаки - 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.

Использовать разделение графов на простые и сложные признаки можно, чтобы более точно моделировать свойства графов. Например, мы можем использовать простые признаки для генерации графов с определенными статистическими характеристиками, такими как размер, плотность, средний путь и т.д. А сложные признаки мы можем использовать для генерации графов, которые имеют определенные свойства, связанные с конкретной областью применения, например, для медицинских молекул - с определенными фармакологическими свойствами.

5 Вычислительный эксперимент

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

5.1 Датасеты

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

5.1.1 Cora

Датасет «Cora» [**sen:aim08**] - 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.

Кроме того, 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.

5.1.2 Citeseer

В качестве второго датасета для тестирования нашего метода был выбран датасет «Citeseer» [16], 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.

5.2 Постановка задачи

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 это задачи с маркированными ребрами в матрице смежности графа, которая может быть описана следующим образом.

Дано:

Граф G с n вершинами и матрицей смежности A размера (n, n), где $A_{ij} = 1$, если ребро (i, j) существует в графе, и 0 в противном случае. Набор маркированных ребер $E = (i, j)$ существующих в графе.

Задача:

Предсказать вероятность существования ребер между всеми парами вершин в графе, включая немаркированные ребра. Однако особый интерес представляет предсказание наличия маркированных ребер. Формально, задача может быть сформулирована как задача бинарной классификации для каждой пары вершин. 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.

Модель машинного обучения обучается на обучающей выборке, которая состоит из пар вершин с маркированными и немаркированными ребрами. Она должна определить, какие признаки графа могут помочь в предсказании наличия ребер, и на основе этих признаков 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, а также предсказание наличия маркированных ребер.

5.3 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, в ряде статей [8] было показано, что необходимо нормализовать матрицу смежности. Причем наилучший способ нормировки выглядит следующим образом:

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

- 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 прохода блоком GCN из получившихся признаков вершин вычитался, преобразованный в скрытое пространство, вектор простых статистик. Особенность скрытого пространства простых статистик заключается в том, что мы требуем, чтобы оно было как можно сильнее похоже на скрытое представление графа. Для этого мы рассчитываем среднее MSE между признаками вершины и преобразованным вектором простых статистик и при обратном проходе градиента, стремимся его уменьшить. Тем самым с каждой итерацией обучения алгоритма мы приближаем скрытый вектор простых статистик к вектору смешанных статистик. Таким образом, в скрытом векторе графа останутся только та информация, которую нельзя получить при помощи простых статистик, то есть вектор сложных статистик.

Теперь для получения скрытого вектора 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 прибавляем к нему преобразованные в скрытое представление простые статистики и создаем матрицу смежности, где в ячейке матрицы находится вероятность того, что на этом месте должно быть ребро.

На каждой эпохе часть ребер случайным образом выбиралось и маркировалось. 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 выбранных ребрах и на всей матрицы смежности в целом, кроме того прибавлялась KL-дивергенция между распределением нашего скрытого вектора и

стандартным нормальных распределением и MSELoss с этапа приближения простых признаков.

Таким образом 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.

5.4 Результаты

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

Однако для подсчета разницы между статистиками входного и сгенерированного графов, нам необходим бинаризовать сгенерированную матрицу. Для это, 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 MAE между простыми статистиками входного графа и сгенерированного.

Такой минимум мы считаем с лучшим результатом, который может получить алгоритм и генерации и его записываем в таблицу результатов 5.4.

	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 5.4, предложенный метод генерирует графы с статистиками, более близкими к исходным, чем обычная модель GraphVAE.

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.

6 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 актуальность 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, что делает наш метод важным шагов в развитии методов контролируемой генерации.

Область использования предложенного подхода обширна, поскольку позволяет учитывать контекст прикладной задачи и явно корректировать генерацию новых сеплов данных. Это может быть полезным во многих приложениях, таких как генерация молекулярных структур или сетей связей между людьми.

Кроме того, в будущих работах мы планируем расширить предложенный метод используя дополнительно новые простые статистики. Также мы планируем исследовать влияние различных простые статистик на ограничение свободы генерации графов. В дальнейшем мы будем исследовать различные методы агрегации графов в векторе одной вершины, что может помочь с более точной генерацией и классификацией графов.

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

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

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

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

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

[16] 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.