Санкт-Петербургский Государственный Университет
Прикладной математики - Процессов Управления

Кафедра Технологии Программирования

Курапов Александр Александрович

Методы обработки, прогнозирования и оценки структуры сложных сетей. Анализ существующих и разработка новых на примере веб-графов

Бакалаврская работа

Научный руководитель:
Кандидат физико-математических наук Сергеев С. Л.

Санкт-Петербург
2021

Оглавление

Введение

Исследование свойств больших сетей является на данный момент актуальной задачей не только в мире математики, но и в других областях науки.

В настоящее время размеры сетевых структур могут достигать несколько миллионов элементов, и тогда возникает задача об оптимальном анализе свойств сети в условиях ограниченности времени и вычислительных мощностей. Существует множество известных алгоритмов для вычисления мер сети (кратчайших путей, центральности, плотности, степени узлов, коэффициентов кластеризации и т.д.). Но некоторые из них становятся непрактичными для больших графов. Таким образом, встаёт вопрос возможно ли получить репрезентативный подграф, меньший по размеру, но с сохранением свойств изначального графа? Если это так, то запуск алгоритма на преобразованном графе будет иметь тот же эффект, что и на исходном. Кроме того, мы можем оценить характеристики исходного графа, используя только подграф.

Для того, чтобы уменьшить размер графа, и одновременно сохранить его свойства, существует множество сложных подходов. Например, можно сформулировать задачу математического программирования, чтобы минимизировать расстояние между исходным и выборочным графом. Этот подход, будучи очень точным, может быть чрезвычайно дорогостоящим. Например, решение задачи уменьшения размера графа таким образом, чтобы разрезы сохранялись, является NP-трудной [2]. Другим недостатком этих подходов является то, что они обычно требуют полной информации (всего графа). Это делает его бесполезным в некоторых сценариях, например в децентрализованных социальных сетях (DSN), где мы можем получить только часть данных и предположить, что это выборка (по некоторому распределению) из исходного графа.

Простой, но эффективный способ преобразования графа заключается в построении выборки (Graph Sampling): выбрать подмножество вершин или рёбер исходного графа. Самым большим преимуществом методов выборки является их эффективность выполнения, так что процедура преобразования не займет больше времени, чем простое вычисление на исходном графе.

В данной работе будут рассмотрены основные характеристики и существующие методы для обработки, прогнозирования и оценки структуры сложных сетей. Основной задачей будет являться выявление новых зависимостей и способов для решения проблемы анализа веб- графов в сети.

Постановка задачи

Целью работы является изучить существующие методы и алгоритмы, которые могут уменьшить размер графа таким образом, чтобы основные свойства и характеристики были сохранены. Также в процессе работы требуется сравнить эти алгоритмы, выявить какие из них работают лучше и при каких обстоятельствах. Помимо этого стоит цель предложить новые методы или подходы и сравнить с существующими алгоритмами.

Для достижения этих целей были поставлены следующие **задачи**:

1. Изучить существующие статьи и описанные в них методы и подходы.

2. Реализовать построение веб-графа, а также методов, предложенных в изученных статьях.

3. Предложить свой метод к извлечению эффективных выборок.

4. Предложить подход к сравнению эффективности методов.

5. Сравнить полученные в ходе работы методы на основании предложенного подхода. Датасетом будут являться настоящие веб-графы реальных социальных сетей.

1. Описание предметной области

1.1. Актуальность работы

Для начала, стоит затронуть тему актуальности и применимости методов выборки в нескольких конкретных примерах:

• **Отсутствие (потеря) данных.** В связи с возможным ограничением вызовов API сайта, бывает невозможно извлечь всех людей из онлайн социальной сети . Вместо этого случайным образом отбирается список индивидов, и некоторые связи(ребра) теряются в процессе. Полезно знать, насколько хорошо этот процесс сохраняет определенные свойства графа. Или, например, сколько вершин / рёбер нужно отобрать, чтобы получить достойную оценку определённой информации?

• **Анонимный опрос населения.** В социологических исследованиях, очень часто нужно сделать опрос и узнать, сколько людей подвержены, например, психологическим расстройствам. Как правило, невозможно непосредственно перечислить и вычислить таких людей. Исследователи обычно начинают с небольшой выборки и расширяют результаты в соответствии с полученными знаниями.

• **Уменьшение стоимости испытаний.** Например, Protein Interaction Network [8] является частым объектом изучения в биохимических исследованиях. Точная проверка взаимодействий между всеми возможными соседями может оказаться слишком дорогостоящей. В этом случае нужно взять часть графа и проверить ребра только на нём, обычно выборка даёт простое решение.

• **Визуализация.** Исходный граф может быть слишком большим, чтобы поместиться на экране. Отображение всех рёбер может быть слишком загроможденным. Выборка может стать решением, что облегчит визуализацию [7].

Все эти примеры демонстрируют широкую применимость методов оценки структуры больших сетей.

1.2. Обзор литературы

В приложенной литературе рассматриваются исследования об извлечении эффективных выборок.

В статье "A Survey and Taxonomy of Graph Sampling" [11] авторы определяют что такое граф, его метрики, характеристики, дают общий обзор на некоторые существующие методы, определяют, что такое Graph Sampling и где это применимо.

В статье "Subnets of scale-free networks are not scale-free: sampling properties of networks" [8] авторы применяют методы случайной выборки на практике и показывают, как и при каких обстоятельствах большие сети могут отклоняться от степенного закона.

В статье "Sampling from large graphs" [4] авторы провели собственный анализ методов извлечения подграфов, предложили новые идеи для измерения подобия таких графов и предложили свой метод Forest Fire.

В статье "On Random Walk Based Graph Sampling"[12] авторы более детально исследовали методы случайных блужданий и предложили свои идеи для модификации этих алгоритмов.

Остальные приложенные статьи являются либо обзорами различных алгоритмов, либо первоисточниками существующих методов.

2. Известные алгоритмы

В данной главе описаны существующие методы Graph Sampling, их сравнительная характеристика, недостатки и преимущества.

2.1. Методы случайной выборки по вершинам

Узловая выборка(Node Sampling): в основе всех этих алгоритмов лежит подбор вершин из исходного графа на основании некоторой топологической информации или в соответствии с неким распределением. Отобранные вершины будут являться вершинами нового подграфа, в том числе рёбра, соединявшие их в исходном, будут соединять их и в меньшем подграфе.

Random Node(Случайная узловая выборка)[8]: процесс выборки в данном случае выглядит так: каждый узел $v^{(i)}$ в исходной сети $\mathcal{N}$ будет включен в подсеть $\mathcal{S}$ с вероятностью p и не включен с вероятностью $(1-p)$. Для конечных графов ожидаемый размер подграфа будет равен $E[X] = Np$ с дисперсией $D[X] = Np(1-p)$. Было доказано, что в безмасштабных(масштабно-инвариантных) сетях распределение степеней отобранного подграфа не соответсвует распределению в исходном.

В качестве распределения степеней можно использовать не только постоянную величину. Исследователи также предложили алгоритмы, в которых распределение вероятности изъятия вершин было бы пропорционально отношению показателя степени вершины к сумме всех степеней (Random Degree Node(узловая выборка на основе степеней))[4] или же некой другой характеристикой, например PageRank[13, 4].

Таблица 1: Сравнение методов, размер выборки : 50%

	Facebook				**LastFm**				**Deezer**			
	Degree connect	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**
RN	0,205	0,186	0,169	0,326	0,208	0,210	0,246	0,367	0,313	0,266	0,217	0,445
RPN	0,047	0,042	0,082	0,107	0,070	0,082	0,063	0,116	0,126	0,076	0,079	0,230
PDN	0,175	0,115	0,170	0,040	0,151	0,171	0,111	0,086	0,055	0,036	0,022	0,148

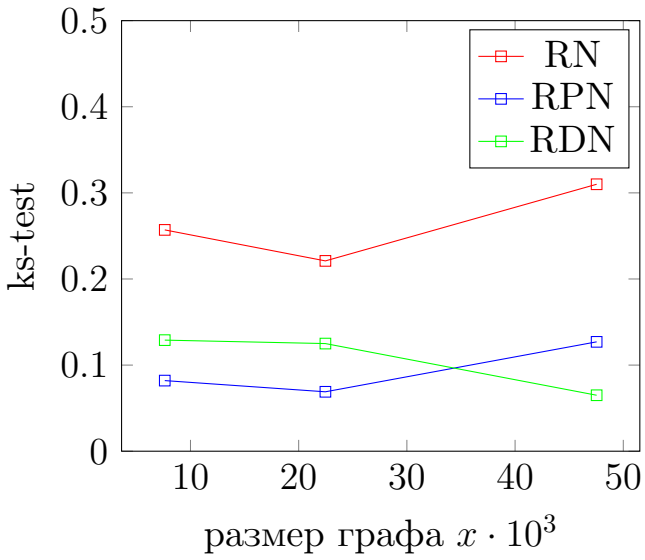
Таблица 2: Сравнение методов, размер выборки : 25%

	Facebook				LastFm				Deezer			
	Degree connect	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**
RN	0,396	0,384	0,410	0,601	0,428	0,394	0,787	0,635	0,565	0,505	0,522	0,703
RPN	0,077	0,066	0,135	0,277	0,107	0,072	0,135	0,360	0,360	0,293	0,280	0,469
PDN	0,166	0,122	0,236	0,148	0,126	0,207	0,158	0,266	0,268	0,190	0,159	0,386

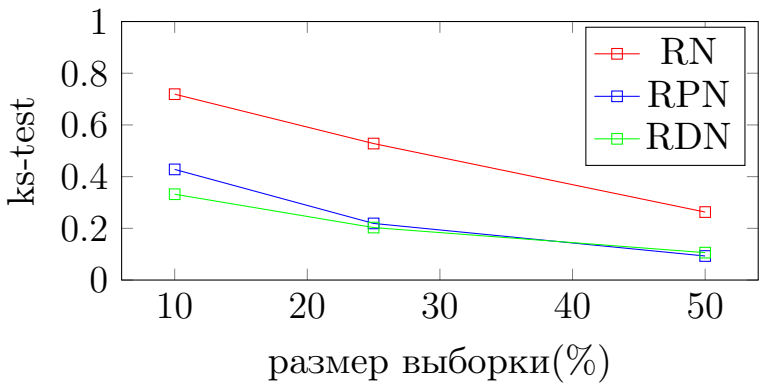
Таблица 3: Сравнение методов, размер выборки : 10%

	Facebook				LastFm				Deezer			
	Degree connect	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**
RN	0,574	0,565	0,642	0,789	0,608	0,496	0,969	0,806	0,773	0,662	0,890	0,864
RPN	0,228	0,212	0,260	0,557	0,190	0,106	0,476	0,620	0,627	0,553	0,607	0,703
PDN	0,080	0,111	0,259	0,351	0,145	0,134	0,198	0,554	0,552	0,490	0,679	0,436

Зависимость значений ks-test от размера изначального графа при размере выборки в 50%



Зависимость значений ks-test от размера выборки



2.2. Методы случайной выборки по рёбрам

На подобии подбора вершин в соответствии с некоторым распределением, аналогично можно подбирать и рёбра. Random Edge(Случайная рёберная выборка)[15] преследует ту же идею, что и Random Node, каждое ребро $w^{(i)}$ в исходной сети $\mathcal{N}$ будет включен в подсеть $\mathcal{S}$ с вероятностью p и не включен с вероятностью $(1-p)$. С этой идеей связано несколько проблем: выборки будут очень слабо связаны и, следовательно, будут иметь большой диаметр и не будут корректно отображать структуру графа.

Интуитивно понятно, что выборка, полученная методом Random Edge, будет слегка предвзята к узлам высокой степени, так как они имеют больше ребер, инцидентных к нему. Эту проблему решает алгоритм Random Node-Edge(Случайная выборка рёбер-вершин)[15], где мы сначала равномерно случайным образом выбираем узел $v^{(i)}$, затем равномерно случайно выбираем ребро $w^{(i_1)}$, инцидентное узлу, и добавляем его к новому графу. В таком случае предвзятость к узлам высокой степени пропадает.

Объединив обе эти идеи извлечения подграфа, был предложен метод, в котором на каждой итерации с вероятностью p добавляется ребро, полученное алгоритмом RNE или с вероятностью $(1-p)$ добавляется ребро, полученное методом RE. Было доказано, что при $p = 0.8$ этот алгоритм работает лучше всего[15].

Таблица 4: Сравнение методов, размер выборки : 50%

	Facebook				**LastFm**				**Deezer**			
	Degree connect	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**
RE	0,159	0,323	0,049	0,247	0,150	0,254	0,112	0,316	0,304	0,364	0,112	0,346
RNE	0,179	0,300	0,292	0,454	0,184	0,322	0,051	0,433	0,366	0,385	0,604	0,641
HYB	0,190	0,330	0,129	0,420	0,202	0,337	0,046	0,382	0,375	0,418	0,167	0,580

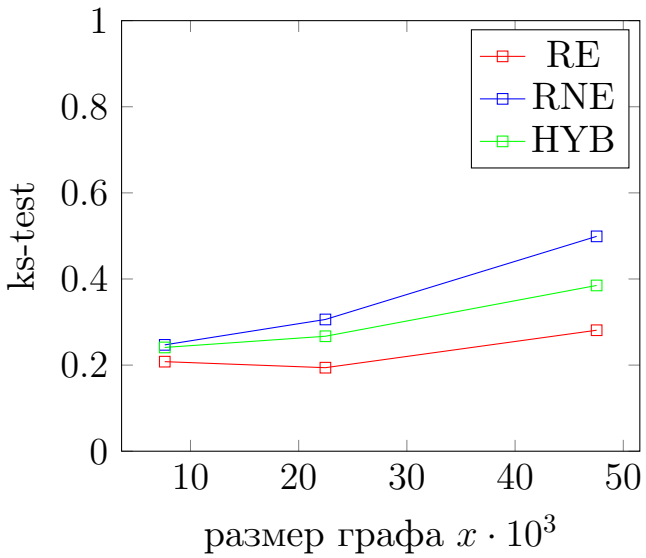
Таблица 5: Сравнение методов, размер выборки : 25%

	Facebook				LastFm				Deezer			
	Degree connect	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**
RE	0,303	0,520	0,117	0,486	0,284	0,430	0,294	0,531	0,536	0,612	0,288	0,639
RNE	0,383	0,506	0,083	0,634	0,376	0,515	0,755	0,616	0,616	0,646	0,687	0,777
HYB	0,381	0,535	0,038	0,578	0,364	0,504	0,630	0,596	0,608	0,648	0,312	0,8

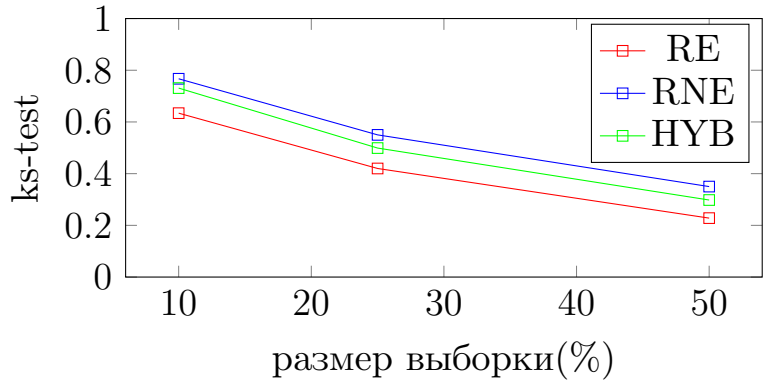
Таблица 6: Сравнение методов, размер выборки : 10%

	Facebook				LastFm				Deezer			
	Degree connect	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**
RN	0,470	0,663	0,278	0,718	0,439	0,540	0,659	0,765	0,725	0,694	0,840	0,827
RPN	0,594	0,729	0,896	0,776	0,529	0,569	0,990	0,765	0,77	0,698	0,992	0,901
PDN	0,578	0,728	0,421	0,828	0,501	0,564	0,993	0,826	0,762	0,699	0,990	0,888

Зависимость значений ks-test от размера изначального графа при размере выборки в 50%



Зависимость значений ks-test от размера выборки



2.3. Аналитические методы

В этом разделе описаны алгоритмы, основанные на более взвешенном подходе к анализу графа, когда выбор следующей вершины $v^{(i)}$сети основывается на свойствах вершин $\{v^{(0)}...v^{(i-1)}\}$, полученных на предыдущих шагах.

2.3.1. Метод "Снежного кома"

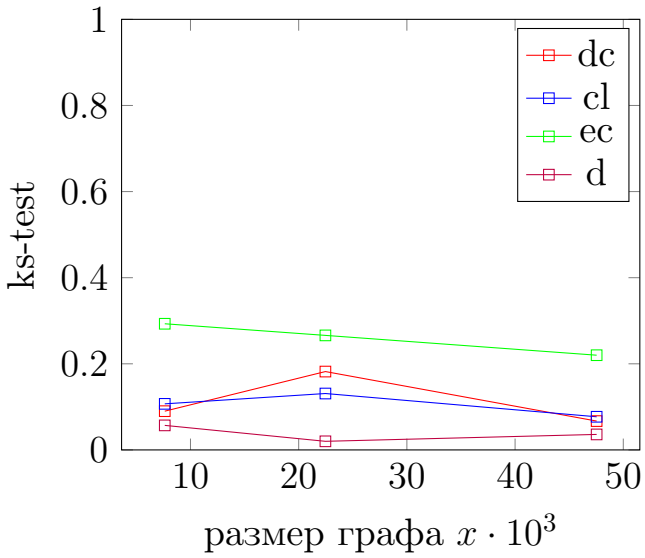
Это метод выборки[5], при котором существующие участники исследования набирают будущих субъектов из числа их знакомых, а знакомые из своих знакомых и т.д. . Выборка методом "Снежного кома" уже давно используется в социологических исследованиях, где проводятся исследование скрытых групп населения (например, алкоголиков). Алгоритм извлечения подграфа выглядит так:

1. Начинаем с некоторого небольшого количества вершин V^o. Эти вершины могут быть получены с помощью случайной выборки по вершинам или как-то иначе.

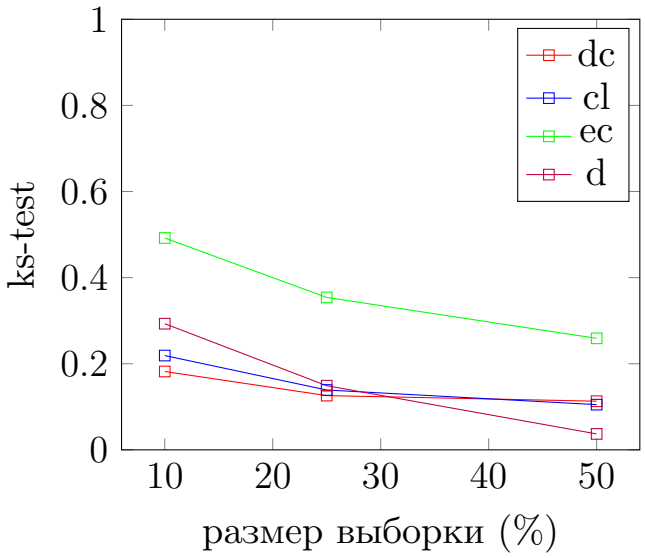
2. На этапе i мы берем $\forall v \in V^{(i-1)}$ k ближайших вершин(соседей). Это эквивалентно взятию выборки инцидентных рёбер к $V^{(i-1)}$, назовем эти рёбра $E^{(i)}$. Вершины, обнаруженные на этом этапе $\bar{V}^{(i)} = \{u, v|(u, v) \in E^{(i)}\}$, добавляем к остальным $V^{(i)} = \bar{V}^{(i)} + \bigcup_{j=0}^{i-1} V^{(j)}$.

3. Процесс повторяется t раз или до тех пор, пока необходимый размер выборки $|E_s|$ не будет достигнут. В результате получаем подграф $G =< V_s, E_s >$, где $V_s = \bigcup_{j=0}^{t} V^{(j)}$ и $E_s = \bigcup_{j=0}^{t} E^{(j)}$.

Зависимость значений ks-test **SB** от размера изначального графа при размере выборки в 50%



Зависимость значений ks-test **SB** от размера выборки



2.3.2. Метод случайных блужданий

Алгоритм Random Walk(RW)[9] выглядит так:

1. Сначала выбираем случайную вершину $v^{(i)}$.

2. Симулируем случайное блуждание по ближайшим соседям, получаем $v^{(i+1)} \in N(v^{(i)})$.

3. Повторяем до тех пор, пока необходимый размер выборки $|E_s|$ не будет достигнут. В результате получаем подграф $G =< V_s, E_s >$.

Стоит отметить, что метод случайных блужданий не может запоминать узлы, которые прошёл. В методе "Снежного кома" вершины

предыдущих этапов исключались из последующего анализа и на них нельзя было вернуться, в RW можно. Это свойство делает его более применимым для теоретического анализа, например получения стационарного распределения из анализа цепей Маркова.

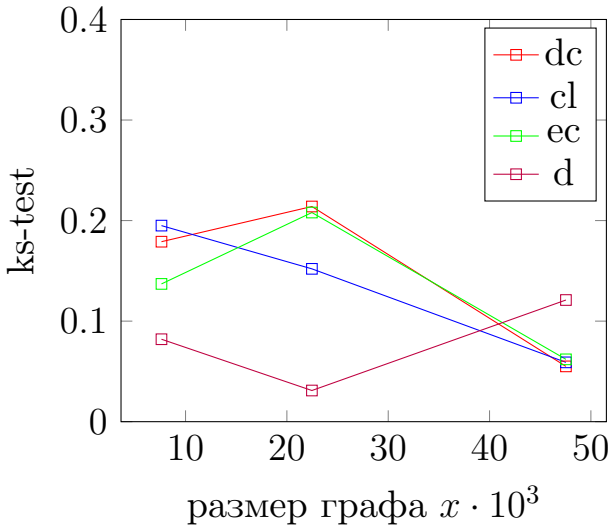
Матрица переходных вероятностей $P(x, y)$ описывает вероятность перехода от узла x к узлу y:

$$P(x,y) = \begin{cases} \frac{1}{degree(x)}, & \text{если y - сосед x.} \\ 0, & \text{иначе.} \end{cases}$$

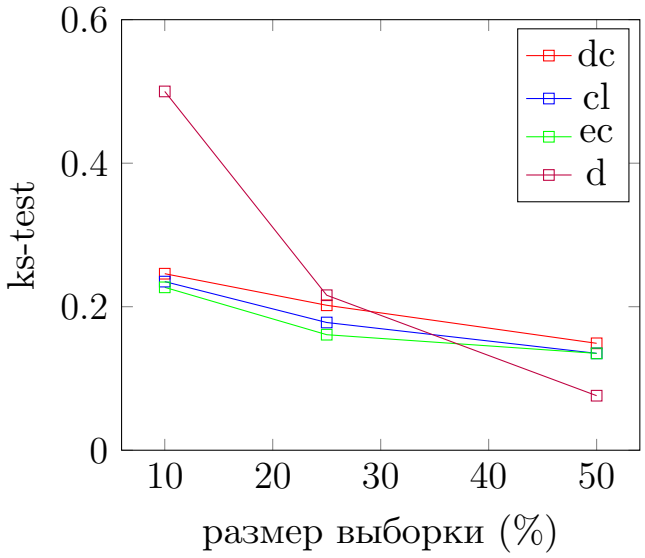
Допустим, вектор v описывает вероятность на текущем узле, тогда вектор $v' = vP$ описывает вероятность после одного перехода. Похожим образом, $v'P^r$ описывает вероятность после r шагов. До тех пор, пока граф связный и не двудольный, вероятность нахождения в любом узле, сходится к стационарному распределению:

$$\pi(x) = \lim_{x\to\infty}(vP^r)(x) = \frac{degree(x)}{2\cdot|E|}$$

Зависимость значений **RW** ks-test от размера изначального графа при размере выборки в 50%

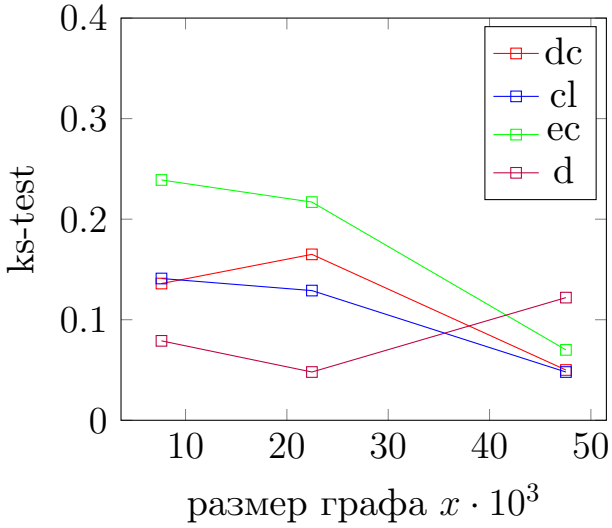


Зависимость значений ks-test **RW** от размера выборки

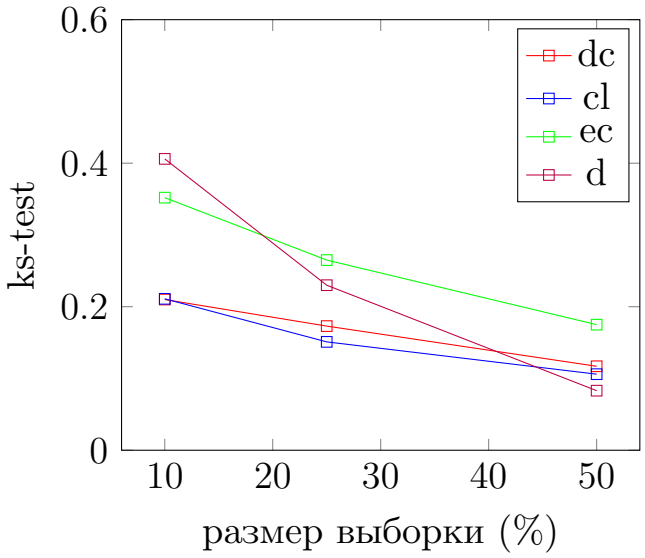


Также было предложено[4] после каждого случайного шага, с некоторой небольшой вероятностью возвращаться в начальную позицию и начинать случайные блуждания заново(Random Walk with Restart).

Зависимость значений **RWR** ks-test от размера изначального графа при размере выборки в 50%

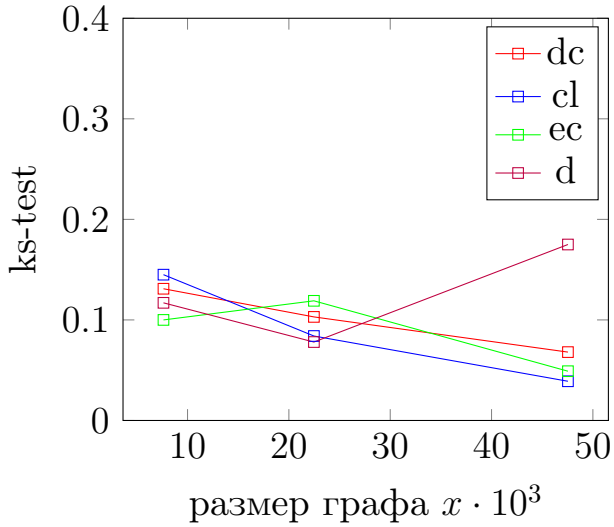


Зависимость значений ks-test **RWR** от размера выборки

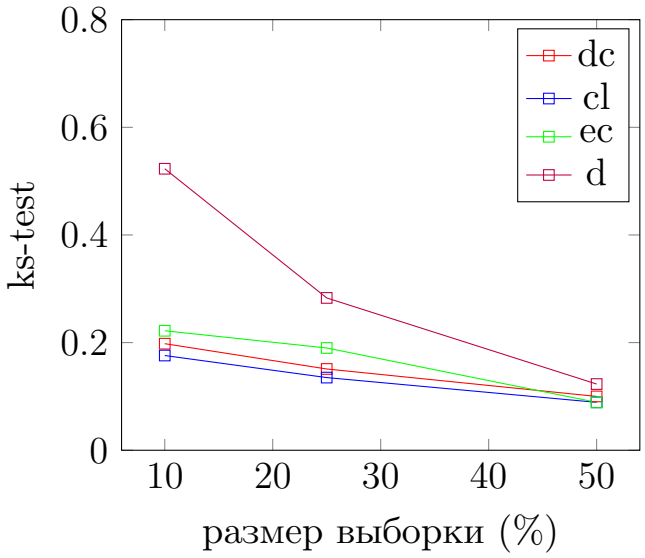


Существует ещё одна широко используемая имплементация случайных блужданий, когда помимо случайного шага к соседу, есть вероятность перейти к произвольному случайному узлу в V. Эта техника называется Random Walk with Jump[4, 6] и она позволяет избавиться от проблемы, когда случайные блуждания по графу приводят в тупик, и дальнейшее продвижение по вершинам происходит очень локально по одним и тем же узлам.

Зависимость значений **RWJ** ks-test от размера изначального графа при размере выборки в 50%



Зависимость значений ks-test **RWJ** от размера выборки



2.3.3. Алгоритм Метрополиса-Хастингса

Алгоритм Метрополиса-Хастингса[10] широко использовался в цепи Маркова Монте-Карло для получения желаемого распределения вершин из произвольного неориентированного связного графа. В методе классических случайных блужданий матрица переходных вероятностей $P(x, y)$ стремится к страционарному распределению $\pi(x)$, как описано выше. Выберем новую матрицу переходных вероятностей $Q(x, y)$, чтобы получить другое распределение $\mu(x)$. В частности, требуется получить равномерное распределение $\mu(x)$, чтобы к концу случайных блужданий с одинаковой вероятностью обойти все узлы. Алгоритм Метрополиса-Хастингса[10, 14]предлагает представить $Q(x, y)$ так:

$$Q(x,y) = \begin{cases} P(x,y)\min(\frac{\mu(y)P(y,x)}{\mu(y)P(x,y)}, 1), & \text{если x} \neq \text{y.} \\ 1 - \sum_{z \neq x} Q(x,z), & \text{если x} = \text{y.} \end{cases}$$

Эквивалентно, будем делать шаг из узла x, выбирать соседа y (с вероятностью $P(x, y)$), затем с вероятностью $\min(\frac{\mu(y)P(y,x)}{\mu(y)P(x,y)}, 1)$ принимать переход в y. Иначе, возвращаемся в узел x(с вероятностью $1 - \sum_{z \neq x} Q(x, z)$). Для того, чтобы делать переходы с одинаковыми вероятностями, требуется $\frac{\mu(y)}{\mu(x} = 1. \Rightarrow$

$$\min(\frac{\mu(y)P(y,x)}{\mu(y)P(x,y)}, 1) = \min(\frac{degree(x)}{degree(y)}, 1)$$

Тогда алгоритм принимает вид:

1. Выбираем у узла x соседа y.

2. Считаем степень узла y.

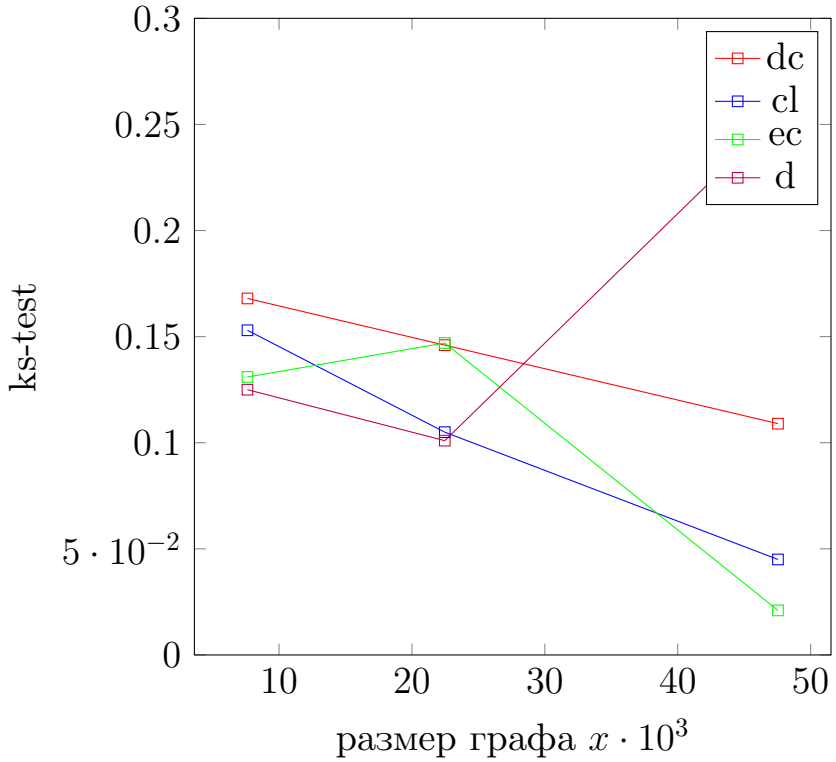
3. Генерируем значение $p \in (0, 1)$.

4. Если $p \leq \frac{degree(x)}{degree(y)}$, выбираем y в качестве узла для следующего этапа.

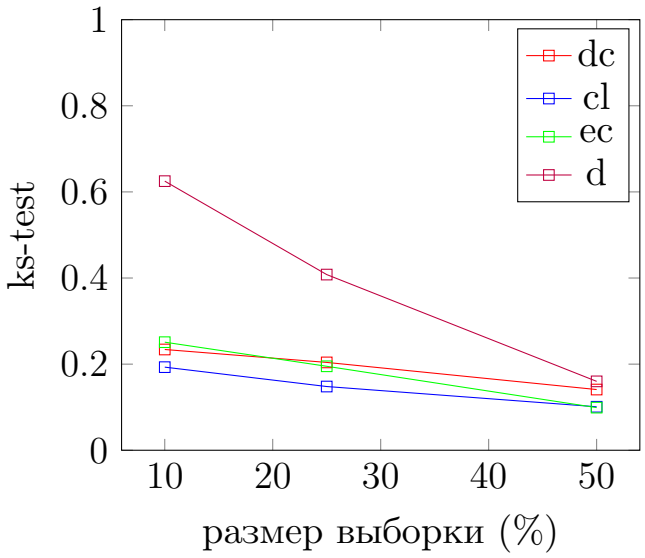
5. Иначе, оставляем x.

Такой алгоритм существенно избавляется от предъвзятости к выбору узлов с высокой степенью, что приводит к подбору каждого узла с равной вероятностью, однако есть и свои минусы. Чтобы выбирать узлы равномерно, MHRW отклоняет значительное количество узлов, которые возможно имели бы больший вес ценность. Следовательно, этот алгоритм страдает от проблемы отклонения узлов.

Зависимость значений **MHRW** ks-test от размера изначального графа при размере выборки в 50%



Зависимость значений ks-test **MHRW** от размера выборки



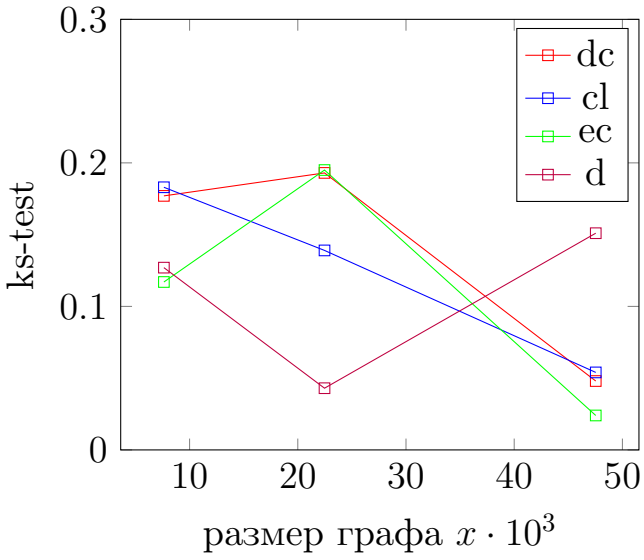
2.3.4. Алгоритм Метрополиса-Хастингса с ограничением

В работе [12] было предложено в алгоритм Метрополиса-Хастингса добавить значение α, с помощью которого можно управлять отклонением или принятием нового узла. Матрица переходных вероятностей $Q(x, y)$ тогда будет выглядеть так:

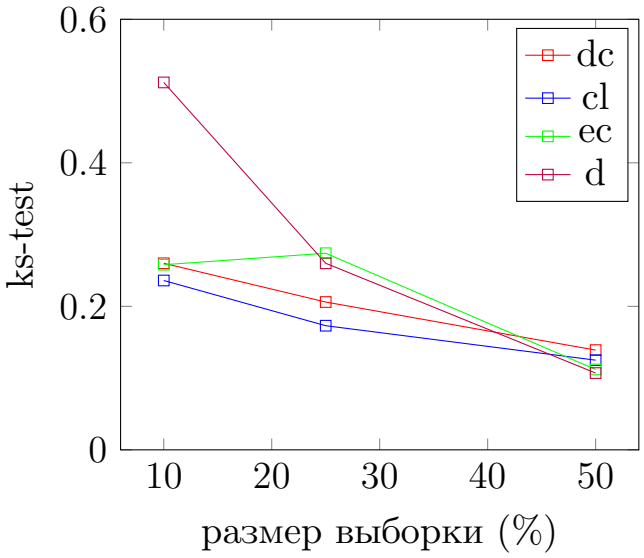
$$Q(x,y) = \begin{cases} P(x,y)\min((\frac{\mu(y)P(y,x)}{\mu(y)P(x,y)})^{\alpha}, 1), & \text{если x} \neq \text{y.} \\ 1 - \sum_{z \neq x} Q(x,z), & \text{если x} = \text{y.} \\ 0, & \text{иначе} \end{cases}$$

В результате RC-MHRW может достичь компромисса между проблемой большого предъвзятости RW к вершинам с высокими степенями и проблемой отклонения узлов MHRW. Кроме того, когда $\alpha = 1$, алгоритм 1 становится алгоритмом MHRW, и когда $\alpha = 0$, алгоритм 1 становится алгоритмом RW. Таким образом, алгоритм 1 создает интересную связь между MHRW и RW, а также объединяет их. Было доказано[12], значение α стоит выбирать в промежутке [0, 0.3].

Зависимость значений **RC-MHRW** ks-test от размера изначального графа при размере выборки в 50%



Зависимость значений ks-test **RC-MHRW** от размера выборки



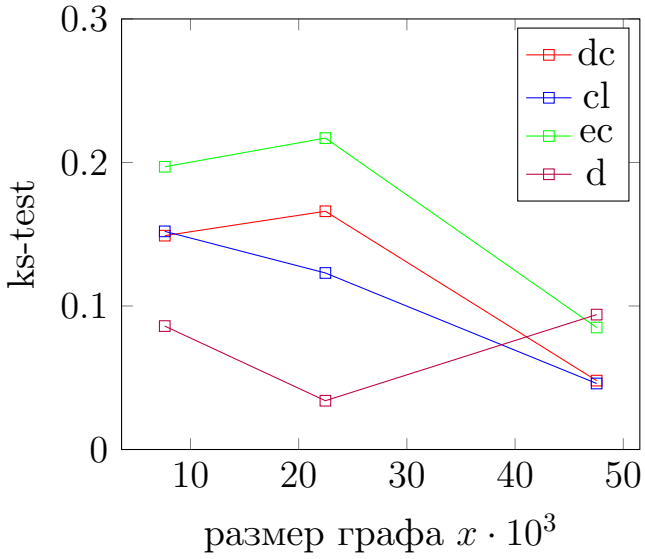
2.3.5. Forest Fire

Алгоритм Forest Fire впервые был представлен в 2005 году[3] в качестве модели генерации графов, которая фиксирует некоторые важные наблюдения в реальных социальных сетях, такие как закон уплотнения, уменьшение диаметра и привязка к обществу. В [4] автор адаптировал эту модель генерации графов для выполнении выборки графов и назвал её Forest Fire Sampling.

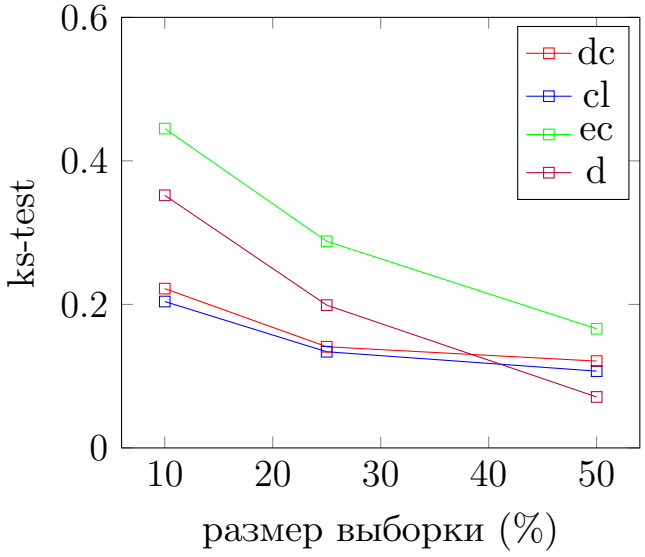
FF - это вероятностная модель алгоритма ”снежного кома”. Начинаем с некоторого узла $v^{(i)}$, затем генерируем число соседей $k^i \approx$

$Geometric(p)$[1], где $P(X = k) = p(1-p)^k$. Узел $v^{(i)}$ выбирает k^i инцидентных узлов, которые ещё не посетил, и добавляет их в подграф. Пусть $w_1, w_2, ..., w_k$ обозначают эти узлы. Затем мы рекурсивно применяем этот шаг к каждому из $w_1, w_2, ..., w_k$, пока необходимый размер выборки $|E_s|$ не будет достигнут. Этот алгоритм был поэтому и назван Forest Fire, потому что после каждого перехода из узла $v^{(i)}$, мы "сжигаем" его и больше не можем достигнуть, эта особенность и выделяет FF из остальных моделей случайных блужданий.

Зависимость значений **FF** ks-test от размера изначального графа при размере выборки в 50%



Зависимость значений ks-test **FF** от размера выборки



3. Эксперимент

В Главе 2 были рассмотрены методы и алгоритмы построения подграфа большой сети для последующей обработки, прогнозирования и оценки структуры графа. Для оценки качества методов был поставлен эксперимент, а также был предложен собственный алгоритм. По результатам поставленного эксперимента сравнивается эффективность каждого из алгоритмов и выявляется наилучший.

3.1. Модификация метода случайного обхода

В качестве основы нового метода возьмем классический Random Edge, когда каждое ребро $w^{(i)}$ добавляется в новый подграф с некой вероятностью. Вместо того, чтобы генерировать вероятность попадания каждого ребра в подграф, будем добавлять ребро $w^{(i)}$, только в том случае, если инцидентные к ней узлы $v^{(i1)}$ и $v^{(i2)}$ уже присутствуют в подграфе. Иначе генерируем число $\alpha \in (0, 1)$, если α окажется меньше заранее заданного значения p, то ребро добавляется в подграф, в ином случае нет.

На практике было доказано, что наилучшее значение $p = 0,1$. Это связано с тем, что в таком случае итоговый подграф, который мы получаем в среднем имеет в 2-3 раза меньше ребер и в 2 - 2.5 раза меньше узлов(в классическом Random Edge количество узлов в новом подграфе составляло 80-90% от исходного). По эффективности такой метод не уступает многим представленным выше методам, результаты сравнения представлены в главе 3.3 .

Таблица 7: Показатели ks-test RE*

	Facebook				**LastFm**				**Deezer**			
	Degree connect	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**
RE*	0,028	0,117	0,072	0,100	0,042	0,057	0,108	0,264	0,289	0,303	0,201	0,294

3.2. Описание эксперимента

В эксперименте оценивались характеристики нового веб-графа против исходного. В качестве основных характеристик для оценивания эффективности алгоритмов использовались следующие (будем рассматривать их как распределения для более взвешенного сравнения):

1. Распределение степени связности в графе: для каждого узла $v^{(i)}$ считаем степень связности C_d^i как : $C_d^i = deg(v^{(i)})$ и составляем множество $\{C_d^i\}$.

2. Распределение коэффициентов кластеризации: для каждого узла $v^{(i)}$ считаем коэффициент кластеризации C_i:

$C_i = \frac{|\{e_{jk}:v_j,v_k \in N_i, e_{jk} \in E\}|}{k_i(k_i-1)}$. и составляем множество $\{C^i\}$.

3. Распределение степеней влиятельности(центральности): пусть $A = (a_{v,t})$ - матрица смежности, то есть $a_{v,t} = 1$, если вершина $v^{(i)}$ связана с вершиной t, и $a_{v,t} = 0$ в противном случае. Показатель x центральности вершины $v^{(i)}$ можно определить как:

$x_v = \frac{1}{\lambda} \sum_{t \in M(v)} x_t = \frac{1}{\lambda} \sum_{t \in G} a_{v,t} x_t$, где $M(v)$ представляет соьой множество соседей вершины v, а λ - константа. После преобразований это выражение можно переписать в векторных обозначениях как уравнение для собственного вектора $\mathbf{Ax} = \lambda \mathbf{x}$ и составляем множество $\{x_i^v\}$.

4. Распределение степеней в графе : для каждой степени d считаем количество вершин n^d со степенью d и формируем множество $\{n^d : \forall d\}$. Это множество показывает частоту появлений степеней вершин графа(другими словами, это частотное распределение).

Все рассмотренные методы были реализованы на языке программирования Python[1]. Для сравнения эффективности алгоритма выборки подграфа с исходным графом и оценки её эффективности будем сравнивать распределения с помощью критерия Колмогорова-Смирнова(ks-test), как $D = \max |F'(x) - F(x)|$, где $F'(x)$ и $F(x)$ - распределения характеристик двух графов.

В качестве датасетов были выбраны реальные наборы данных социальных сетей и веб-графов для сравнения производительности процедур выборки и проверки их полезности для ускорения задач обработки,

прогнозирование и оценки:

1. Facebook - веб-граф официальной страницы Facebook. Узлы - это страницы, представляющие политиков, правительственные организации, ТВ-шоу и компания, рёбра - ссылки между страницами (22,470 узлов и 171,002 ребер).

2. LastFM - социальная сеть. Узлы - пользователи, рёбра - взаимные подписчики (7,624 узлов и 27,806 ребер).

3. Deezer - социальная сеть в Венгрии, узлы - венгерские пользователи, рёбра - дружеские отношения (47,538 узлов и 222,887 ребер).

3.3. Результаты эксперимента

В течение эксперимента каждый из методов был протестирован на трёх датасетах с объемами выборки в 50, 25 и 10 процентов. На таблице ниже представлены результаты эксперимента при объеме выборки в 50 процентов (50% вершин или 50 % рёбер для методов RE, RNE, HYB, RE*), жирным шрифтом выделены наилучшие показатели по каждому из характеристик, полученные в результате эксперимента.

Таблица 8: Сравнение всех методов, размер выборки : 50%

	Facebook				**LastFm**				**Deezer**			
	Degree connect	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**	**Degree connect**	**Clust**	**Eigen connect**	**Degree**
RN	0,205	0,186	0,169	0,326	0,208	0,210	0,246	0,367	0,313	0,266	0,217	0,445
RPN	**0,047**	**0,042**	0,082	0,107	**0,070**	**0,082**	0,063	0,116	0,126	0,076	0,079	0,230
PDN	0,175	0,115	0,170	**0,040**	0,151	0,171	0,111	**0,086**	**0,055**	**0,036**	**0,022**	**0,148**
RE	0,159	0,323	**0,049**	0,247	0,150	0,254	0,112	0,316	0,304	0,364	0,112	0,346
RNE	0,179	0,300	0,292	0,454	0,184	0,322	0,051	0,433	0,366	0,385	0,604	0,641
HYB	0,190	0,330	0,129	0,420	0,202	0,337	**0,046**	0,382	0,375	0,418	0,167	0,580
SB	0,182	0,131	0,266	**0,02**	0,090	0,107	0,293	**0,057**	0,067	0,077	0,220	**0,036**
RW	0,214	0,152	0,208	0,031	0,179	0,195	0,137	0.082	0,055	0,059	0,062	0,121
MHRW	0,146	0,105	0,147	0,101	0,168	0,153	0,131	0,125	0,109	0,045	**0,021**	0,254
RCMH	0,193	0,139	0,195	0,043	0,177	0,183	0,117	0,127	0,048	0,054	0,024	0,151
RWJ	0,103	**0,084**	0,119	0,078	0,131	0,145	**0,1**	0,117	0,068	**0,039**	0,049	0,175
RWR	0,165	0,129	0,217	0,048	0,136	0,141	0,239	0,079	0,050	0,048	0,070	0,122
FF	0,166	0,123	0,217	0,034	0,149	0,152	0,197	0,086	**0,048**	0,046	0,085	0,094
RE*	**0,028**	0,117	**0,072**	0,100	**0,042**	**0,057**	0,108	0,264	0,289	0,303	0,201	0,294

Из таблицы 8 можно выделить несколько методов, средний показатель ks-test(для каждой из хакактеристик и каждого датасета) которых на порядок выше, чем у остальных методов. Это метод случайных блужданий с прыжком RWJ (среднее значение - 0.10066), метод случайных вершин на основе характеристики PageRank RPN (среднее значение - 0.0933) и метод FF (среднее значение - 0.116416).

Очевидно, что, чтобы выявить наилучший метод недостаточно найти метод с наилучшими показателями теста Колмогорова-Смирнова, также стоит учесть размер полученной выборки(как ребер, так и вершин). На следующей таблице представлены размеры полученных выборок в сравнении с изначальным графом для каждого датасета среди наилучших методов(и предложенного метода для сравнения), а также средние показатели ks-test всех метрик для каждого датасета:

Таблица 9: Методы RWJ, RPN, FF, RE*

	Facebook			**LastFm**			**Deezer**			
	Вершины	**Рёбра**	**Среднее**	**Вершины**	**Рёбра**	**Среднее**	**Вершины**	**Рёбра**	**Среднее**	**Общее Среднее**
RWJ	0,5	0,605	0,096	0,5	0,620	0,123	0,5	0,449	0,082	0,100
RPN	0,5	0,539	**0,069**	0,5	0,557	**0,082**	0,5	0,389	0,127	**0,093**
FF	0,5	0,698	0,135	0,5	0,624	0,146	0,5	0,501	**0,068**	0,116
RE*	0,543	0,527	0,079	0,387	0,334	0,117	0,542	0,301	0,271	0,156

Из таблицы 9 можно сделать несколько выводов:

1. Наилучшие показатели ks-test показывает метод случайной выборки вершин, где распределение вероятности изъятия вершин пропорционально характеристике PageRank, и количество рёбер и вершин не превышает в среднем половины исходного, что облегчит дальнейшие вычисления. Но есть существенный недостаток: чтобы произвести такую выборку требуется изначально знать характеристику Pagerank для каждой из вершин или же считать её непосредственно, что может отразиться на трудозатратности и времени работы.

2. Предложенный в ходе работы метод RE*, модифицирующий случайную выборку рёбер, выдаёт неплохие результаты только в том случае, если граф транзитивен(хотя бы на 17 процентов, как в случае с датасетом LastFm), и плохо справляется в ином случае. Это можно объяснить тем, что данный метод может включать рёбра в подграф, только когда обе инцидентные вершины уже присутствуют в нем, иначе ребро отбирается с некоторой вероятностью. Тем не менее даже при таком недостатке, он обходит по эффективности многие методы, учитывая средние показатели ks-test и итоговый размер подграфа.

3. В результате данного эксперимента наиболее эффективными методами с точки зрения подобия и размера подграфа являются FF и RWJ, в среднем оба метода при выборке 50% вершин отбирают 58% рёбер, что существенно уменьшает размер итогового подграфа, и неплохо справляются с поставленной задачей.

Заключение

В выпускной квалификационной работе изучены и проанализированы существующие методы и алгоритмы для обработки, прогнозирования и оценки структуры сложных веб-графов, а также предложен собственный метод для извлечения эффективных выборок графа. В том числе получены следующие научные результаты:

1. Изучены существующие статьи и описанные в них методы и подходы.

2. Реализовано построение веб-графов, а также методов, предложенных в изученных статьях.

3. Предложен свой метод к извлечению эффективных выборок.

4. Предложен подход к сравнению методов.

5. Получен результат эксперимента сравнения полученных в ходе работы методы на основании предложенного подхода.

Таким образом были выполнены все поставленные задачи и достигнута цель работы. Результаты данной работы могут быть применимы, например, в анализе социальных сетей(SNA) для моделирования распространения сети, анализа характерных признаков и поведения элементов графа, а также прогнозирования связей в веб-графах.

Список литературы

[1] Github. Реализация методов // Приватный репозиторий, писать на почту al.kurapov@gmail.com. — 2021. — Режим доступа: `https://github.com/akrpv/graphs`.

[2] Harvey N. Graph sparsifiers: A survey. — Proceedings Scientific-Computing presentation slides, 2011.

[3] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. — In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pages 177–187, 2005.

[4] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. — In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006.

[5] L. A. Goodman. Snowball sampling. — The annals of mathematical statistics, 32(1):148–170, 1961.

[6] L. Jin, Y. Chen, P. Hui, C. Ding, T. Wang, A. V. Vasilakos, B. Deng, and X. Li. Albatross sampling: robust and effective hybrid vertex sampling for social graphs. — In Proceedings of the 3rd ACM international workshop on MobiArch, pages 11–16, 2011.

[7] M. Kurant, M. Gjoka, Y. Wang, Z. W. Almquist, C. T. Butts, and A. Markopoulou. Coarse-grained topology estimation via graph sampling. — In Proceedings of ACM SIGCOMM Workshop on Online Social Networks (WOSN) '12, Helsinki, Finland, 2012.

[8] Michael PH Stumpf, Carsten Wiuf, and Robert M May. Subnets of scale-free networks are not scale-free: sampling properties of networks. — Proceedings of the National Academy of Sciences 102, 12 (2005), 4221–4224, 2005.

[9] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. Walking in facebook: A case study of unbiased sampling of osns. — In 2010 Proceedings IEEE Infocom. Ieee, 1–9, 2010.

[10] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. — The journal of chemical physics, 21:1087, 1953.

[11] Pili Hu, Wing Cheong Lau. A Survey and Taxonomy of Graph Sampling. — Department of Information Engineering Chinese University of Hong Kong, 2013.

[12] Rong-Hua Li, Jeffrey Xu Yu, Lu Qin, Rui Mao, and Tan Jin. On random walk based graph sampling. — In 2015 IEEE 31st International Conference on Data Engineering, 2015.

[13] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. — Comput. Netw.ISDN Syst., 30 (1-7), pp. 107–117, 1998.

[14] S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algorithm. — The Americian Statistician, vol. 49, no. 4, pp. 327–335, 1995.

[15] Vaishnavi Krishnamurthy, Michalis Faloutsos, Marek Chrobak, Li Lao, J-H Cui, and Allon G Percus. Reducing large internet topologies for faster simulations. — s. In International Conference on Research in Networking, 2005.