РАЗРАБОТКА АЛГОРИТМОВ ПОИСКА ИЗОМОРФНЫХ ПОДГРАФОВ В ПОМЕЧЕННЫХ ГРАФАХ

Д. И. Денгалёв

ddengalev@gmail.com;
Научный руководитель — В. И. Сарванов, кандидат физико-математических наук, доцент

Работа посвящена проблема поиска изоморфных индуцированных подграфов в помеченных графах. Лучшими алгоритмами для решения данной задачи для больших и разреженных графов являются алгоритмы серии VF [1]. В настоящей работе предложено улучшение алгоритмов VF3 [2] и VF3-light [3] с использованием машинного обучения.

Ключевые слова: граф; алгоритм; изоморфизм; изоморфное вложение; индуцированный подграф; машинное обучение.

ВВЕДЕНИЕ

В задаче поиска изоморфных подграфов необходимо найти изоморфные вложения одного графа (паттерна) в другой (датаграф).

Задачи поиска изоморфных подграфов имеют чрезвычайно широкое применение. Они возникают, в частности, при анализе социальных сетей, графовых структур в биоинформатике, обработке изображений и проектировании микросхем.

Алгоритмы серии «VF» входят в число основных алгоритмов решения этой NP -трудной задачи. Например, алгоритм VF2 включен в качестве солвера в библиотеки NetworkX и Boost.

Алгоритмы серии VF, как и большинство других алгоритмов решения указанной задачи базируется на схеме поиска с возвратом и отсечениями (бэктрекинге). Наряду с отсечениями ключевую роль в таких алгоритмах играет упорядочение вершин паттерна, т.е., порядок в котором вершины паттерна включаются в бэктрекинг. Выбор «хорошего» (в идеале оптимального) упорядочения позволяет повысить производительность алгоритма. В настоящей работе предлагается новый подход к выбору «хорошего» упорядочения, использующий технику машинного обучения.

ЭФФЕКТИВНОЕ УПОРЯДОЧЕНИЕ

Необходимо найти такой порядок вершин $V' = v_1, v_2, .. v_n$, где n – число вершин графа, который эффективно уменьшает время работы программы, задаваемое функцией $F: V' \rightarrow R$. Эффективное уменьшение означет, что время работы программы на этом порядке значительно

меньше, чем среднее время работы программы при случайном порядке вершин.

Введем вектор весов $\omega = (\omega_1, \dots \omega_k)$, который отвечает за то, в какой степени каждое значение в векторе представлений вершин графа будет влиять на ее позицию в упорядочении.

Для первых трех координат вектора представлений возьмем критерии из алгоритма VF3-light:

1. Число ребер, исходящих из рассматриваемой вершины и ведущих вершины, которые уже упорядоченные.
2. Число вершин в датаграфе, которые имеют такую же или большую степень и ту же метку, что и рассматриваемая вершина паттерна.
3. Степень вершины в паттерне

Остальные координаты вектора представления берутся из специальных представлений вершин в графе. Например, Struc2Vec [4], Role2Vec [5].

Таким образом, алгоритм эффективного упорядочения вершин графа заключается в следующем:

1. Строится набор графов, на которых будем подбирать вектор весов ω (обучающая выборка).
2. К каждому паттерну из этого набора применяем алгоритм представления вершин, заранее выбранный для всех графов из набора (например, Role2vec). Полученные представления сохраняем.
3. При помощи генетического алгоритма или алгоритма «имитации отжига» подбираем вектор весов ω. Для этого решается задача минимизации функции F.
4. Используем полученные на шаге 3 веса для эффективного упорядочения вершин других графов. Упорядочиваем вершины паттерна по возрастанию значений скалярного произведения вектора весов ω и вектора представления вершин паттерна.

Полученное эффективное упорядочение может быть использовано в алгоритме VF3 и VF3-light в качестве порядка вершин паттерна в бэктрекинге.

В настоящей работе предлагается следующий эвристический алгоритм для подбора вектора весов ω:

Алгоритм состоит из k шагов. Шаг этого алгоритма состоит в следующем:

- Генерируются q случайных векторов ω. Из них выбирается лучший, т.е. тот, на котором суммарное время работы алгоритма на обучающей выборке минимально.

- Выполняются m итераций для этого шага, на каждой итерации генерируется случайный вектор ω_{random}:

а) Выбирается случайное число s от 0 до n. На s позициях генерируется ненулевое значение, в остальных позициях генерируется нулевое. Вектор ω заменяется на вектор $\omega + \omega_{random}$ тогда и только тогда, когда суммарное время работы алгоритма на обучающей выборке для вектора весов $\omega + \omega_{random}$ меньше, чем для вектора весов ω.

В конце алгоритма выбирается вектор ω, лучший среди векторов, полученных на всех шагах алгоритма. Полученный вектор является локально оптимальным при достаточно большом числе шагов алгоритма.

На практике алгоритм хорошо работает когда в собранном для подбора ω наборе графов и в наборе графов, на котором алгоритм применяется, один и тот же паттерн. Таким, образом неявно определяется эффективное упорядочение для паттерна, которое в дальнейшем может использоваться для задач с другим датаграфом.

РЕЗУЛЬТАТЫ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ

В эксперименте производился подбор векторов ω на одном наборе датаграфов и замерялось время работы программы на другом наборе. Полученные результаты представлены в виде таблицы.

Ускорение времени работы в зависимости от размера датаграфов и выбранного алгоритма

Размер обучающих датаграфов	Размер тестовых датаграфов	Алгоритм для поиска изоморфных подграфов	Ускорение, в процентах
5000	5000	VF3	7.2%
10000	10000	VF3	13.4%
20000	20000	VF3	6.1%
5000	5000	VF3-light	6.7%
10000	10000	VF3-light	23%
20000	20000	VF3-light	13.2%

В таблице представлено среднее ускорение алгоритмов VF3 и VF3-light в зависимости от размера обучающих и тестовых датаграфов.

ЗАКЛЮЧЕНИЕ

В ходе работы проделано следующее:

1. Изучены и проанализированы алгоритмы серии VF и их программные реализации.
2. Проведен анализ алгоритмов специального представления вершин графа, связанного с машинным обучением.

3. Программно реализованы два классических алгоритма эвристической оптимизации. Предложен и программно реализован новый алгоритм эвристической оптимизации.
4. Разработан и программно реализован новый алгоритм решения задачи эффективного упорядочения вершин паттерна. Этот алгоритм использует методы машинного обучения, что представляет собой новый подход к решению данной задачи.
5. Предложены новые модификации алгоритмов VF3 и VF3-light и их программных реализаций, использующие эффективное упорядочение вершин паттерна

Библиографические ссылки

1. *Carletti V., Foggia P., Saggese A., Vento M.* Challenging the time complexity of exact subgraph isomorphism for huge and dense graphs with VF3 // IEEE transactions on pattern analysis and machine intelligence. 2018. Vol. 40, №. 4. P. 804-818
2. *Carletti V., Foggia P., Saggese A., Vento M.* Introducing VF3: A New Algorithm for Subgraph Isomorphism // Graph-Based Representations in Pattern Recognition. 2017.
3. *Carletti V., Foggia P., Greco A., Vento M., Vigilante V.* VF3-Light: A lightweight subgraph isomorphism algorithm and its experimental evaluation // Pattern Recognition Letters. 2019. Vol. 125. P. 591-596,
4. *Ribeiro, L., Saverese, P.H., Figueiredo, D.R.* struc2vec: Learning Node Representations from Structural Identity // *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* 2017. P. 384-394.
5. *Ahmed N., Rossi R., Lee J., Kong X., Willke T., Zhou R., Eldardiry H.* Learning Role-based Graph Embeddings. // *IEEE Transactions on Knowledge and Data Engineering*. 2022. Vol. 34, №. 5. P. 2401-2415.