

Генерация графовых моделей с помощью LLM

Насыров Э.Р.

1 октября 2024 г.

Описание проекта. Графовые модели

Графовая вероятностная модель — вероятностная модель, в которой в виде графа представлены зависимости между случайными величинами. Вершины соответствуют случайным переменным, рёбра — вероятностным взаимосвязям между случайными величинами.

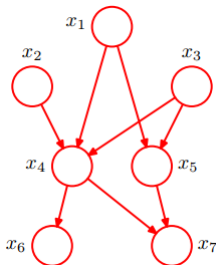


Рис.: Графическая модель.

$$p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3)p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5).$$

Рис.: Совместная плотность.

Описание проекта.

Задача: по описанию ситуации на естественном языке построить соответствующую графическую вероятностную модель.

Архитектура решения.

- 1 Из описания вычленить случайные величины, константы, типы распределений. Они будут вершинами графа.
- 2 Из описания и пользуясь общими знаниями (LLM) выявить вероятностные взаимосвязи. Они будут ориентированными ребрами графа.
- 3 Задать типы распределений случайных величин.
- 4 Написать код на Python (с помощью LLM), задающий созданную графическую модель, пользуясь одной из библиотек (pgmPy, doWhy, PyMC, ...)
- 5 Визуализировать графическую модель.

Описание проекта. Демо

Текстовое описание: У меня есть датасет, в котором указаны возраст человека, наличие у него детей и логарифм времени, которое он изменяет. Я хочу узнать, сколько времени мой муж проводит с любовницей. Составь графическую модель, чтобы я смогла ее обучить и вычислить ответ.



Рис.: Построенная графическая модель по текстовому описанию.

Описание проекта. Сложности

- ❶ **Данные.** Где брать данные для задачи (описание, граф)?
- ❷ **Модель.** Качество open-source LLM ниже, чем закрытых.
Качество 7B-моделей хуже, чем 100B.
- ❸ **Метрики.** Как измерять качество сгенерированных графических моделей?
- ❹ **Лосс.** Какой лосс использовать при обучении?
- ❺ **Неоднозначность.** Одному и тому же текстовому описанию могут соответствовать и разные графы, и разные распределения.

Описание алгоритма.

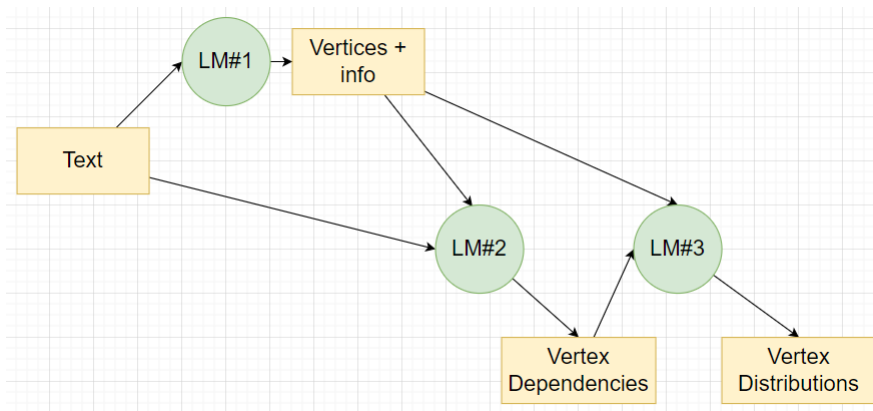


Рис.: Получение характеристик графической модели.

Описание алгоритма.

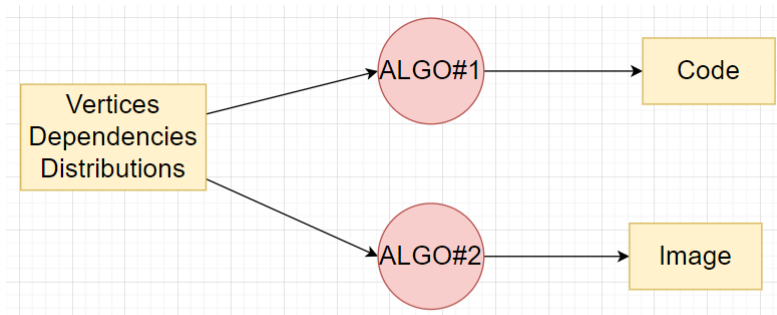


Рис.: Создание ГМ и ее визуализация.

Схема проекта.

Классы:

- 1 **NaturalLanguageInput Class** — отвечает за препроцессинг текста, получение случайных величин, распределений и т.д.
- 2 **GraphModel Class** — структура данных для хранения, создания и визуализации графических моделей. Также используется для валидации (что нет циклов и т.д.).

Список технологий:

- 1 **LLM API:** GPT-4o, o1, Claude 3.5, Gemini
- 2 **Code generation:** LLaMa2, LLaMa3
- 3 **LLM tuning:** Stable LM 2, Mixtral8B, Alpaca
- 4 **Графы:** NetworkX
- 5 **Визуализация:** Graphvis, Matplotlib, Plotly
- 6 **Развертывание:** HF Spaces, Gradio