

Классификация распределения с помощью случайных графов

Соколовский С.П., Григоренко М.Д.

Дата: 16 мая 2025 г.

Предисловие

Договоримся об обозначениях:

- n — размер вектора реализаций случайной величины
- k, d — параметры построения KNN и дистанционного графов соответственно
- θ, v — параметры распределений
- T^{KNN}, T^{dist} — характеристики случайных графов

Часть I. Исследование свойств характеристики

Используемые инструменты Соколовского С.П.

Весь код в ветке `Crazy-Explorer31/first_part`, в директории `src/`:

- `graphs.py` — реализации KNN и дистанционного графов (у каждого есть метод для построения и отрисовки)
- `characteristics.py` — функции для получения характеристик графов, построенных при данных параметрах (распределений, построения графов...). Самый важный — `get_average_characteristics`, возвращающий средние характеристики графов, построенных при переданных параметрах
- `visualisations.py` — функции для удобной построения графиков
- `metrics.py` — функции, приближенно считающие ошибку I рода и мощность для данного \mathcal{A} . Считается по методу Монте-Карло, используя переданное в функцию множество точек (число компонент, кроме числа), принадлежащих какому-то распределению.

Используемые инструменты Григоренко М.Д.

TODO

Шаг 1. Фиксируем n, k, d . Исследуем взаимосвязь между θ, v и T^{KNN}, T^{dist}

Результаты Соколовского С.П.

В файле `experiments_first_part_1.ipynb` происходит следующее:

- Для каждой тройки (распределение, тип графа, характеристика) перебираются параметры трех перечисленных объектов, после чего вычисляются характеристики полученных графов.
- Для каждой тройки строится диаграмма рассеивания, в которой по горизонтальной оси — параметр распределения, а по вертикальной — характеристика графа

Из графиков заметно, что лишь с дистанционным графом хочется продолжать работать

Результаты Григоренко М.Д.

TODO

Шаг 2. Фиксируем θ, v . Исследуем взаимосвязь между n, k, d и T^{KNN}, T^{dist}

Результаты Соколовского С.П.

В файле `experiments_first_part_2.ipynb`, аналогично первому шагу, генерятся много налюдений для всех комбинаций распределений, типов графов, их характеристик. Далее на диаграммах рассеивания по оси Ох откладываются параметры построения графов, по Оу — их характеристики, и ещё цветом отражена, при каком n было получено налюдование. Выводы аналогичные первому эксперименту

Результаты Григоренко М.Д.

TODO

Шаг 3. Фиксируем θ, v . Строим \mathcal{A} для переданного n

Результаты Соколовского С.П.

Файл `experiments_first_part_3.ipynb` поделен на два раздела. В первом фиксируются все параметры и строится \mathcal{A} . Во втором рассуждения, изложенные в первом разделе обобщаются, и приведена реализация класса, строящая \mathcal{A} по переданному в конструктор n .
Используется следующий алгоритм построения \mathcal{A} :

1. Строятся точки с координатами (число компонент, хроматическое число) по генерирующимся векторам случайных величин

2. За изначальное \mathcal{A} берется множество всех сгенерированных точек, полученных по первому распределению (Exp).
3. Далее пытаемся удалить точку из \mathcal{A} так, чтобы ошибка I рода не превысила 0.05, а мощность была максимальной (ошибка I рода и мощность считаются на основе точек, сгенерированных в начале). Для этого перебираем все варианты и выбираем наилучший
4. Пытаемся так удалить что-то из \mathcal{A} много раз
5. В итоге получаем искомое \mathcal{A}

Результаты Григоренко М.Д.

TODO