Классификация распределения с помощью случайных графов

Соколовский С.П., Григоренко М.Д.

Дата: 17 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, хром число), принадлежащих какому-то распределению.

Используемые инструменты Григоренко М.Д.

Весь код в ветке `maxGrigorenko/first_part`, в директории `src/`:

- `graph_common_functions.py` — реализации KNN и дистанционного графов (у каждого есть метод для построения из значений случайной величины, а также методы вычисления характеристик)
- `distribution_functions.py` — функции для генерации выборки и вычисления матожидания характеристики методом Монте-Карло.

Шаг 1. Фиксируем n. 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.

В файле `experiments_first_part_1.ipynb` происходит следующее:

- Реализованы функции `plot_sigma` и `plot_beta`, перебирающие значения соответствующих параметров распределений и выводящих график зависимости характеристики графов (knn и dist) от перебираемого параметра
- При фиксированном размере выборки проведены эксперименты с различными параметрами d и k.

В результате всех экпериментов delta графа knn была константной, то есть эта характеристика никак не связана с параметрами распределений. А вот доминирующее число дистанционного графа в среднем увеличивалось при увеличении параметра sigma.

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 Ox откладываются параметры построения графов, по Oy — их характеристики, и ещё цветом отражена, при каком n было получено наюлюдение. Выводы аналогичные первому эксперименту

Результаты Григоренко М.Д.

В файле `experiments_first_part_2.ipynb` зафиксированы параметры распределений и отрисованы графики зависимости характеристик графов от размера выборки. delta графа knn оказалась неинформативной характеристикой. А вот доминирующее число дистанционного графа немного по-разному меняется при изменении размера выборки, в особенности, если в качестве параметра дистанционного графа установить значение d >= 3, то характеристика графа из нормального распределения становится почти всегда равной 1, а вот при распределении Лапласа немного больше.

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.

В файле `experiments_first_part_3.ipynb` реализован алгоритм конструирования множества $\mathcal{A}$, оно должно содержать такие значения, что при попадании характеристи графа в него, можно было с вероятностью ошибки $\alpha = 0.05$ считать, что распределение является нормальным. Такое множество строилось по результату большого числа экспериментов, если при нормальном распределении значение характеристики графа встречалось хотя бы в $(1 - \alpha)$ доле случаев (то есть доля случаев при распределении Лапласа не более α), то оно добавлялось в множество. Такой критерий оказался довольно слабым, так как множество $\mathcal{A}$ при различных d получалось либо пустым, либо из совсем небольшого числа значений (при большом разбросе значений характеристики). Дополнительно проведены

эксперименты при переставленных распределениях (то есть, если до этого мы пытались по характеристике графа найти значения, по которым можно довольно точно определить, что распределение нормальное, то в данном эксперименте мы пытались найти значения, по которым можно было бы довольно точно определить, что распределение является распределение Лапласа). Здесь результаты оказались лучше, в частности, при фиксированном $d = 3.5$, при различных значениях n, множество оказалось непустым, оно также состояло из 1 или 2 значений, но в данном случае и разброс характеристики совсем небольшой.