

# Классификация распределения с помощью случайных графов

Соколовский С.П., Григоренко М.Д.

Дата: 24 мая 2025 г.

## Предисловие

---

Договоримся об обозначениях:

- $n$  — размер вектора реализаций случайной величины
- $k, d$  — параметры построения KNN и дистанционного графов соответственно
- $\theta, v$  — 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 математического ожидания характеристики методом Монте-Карло.

## Шаг 1. Фиксируем $n$ . Исследуем взаимосвязь между $\theta, v$ 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

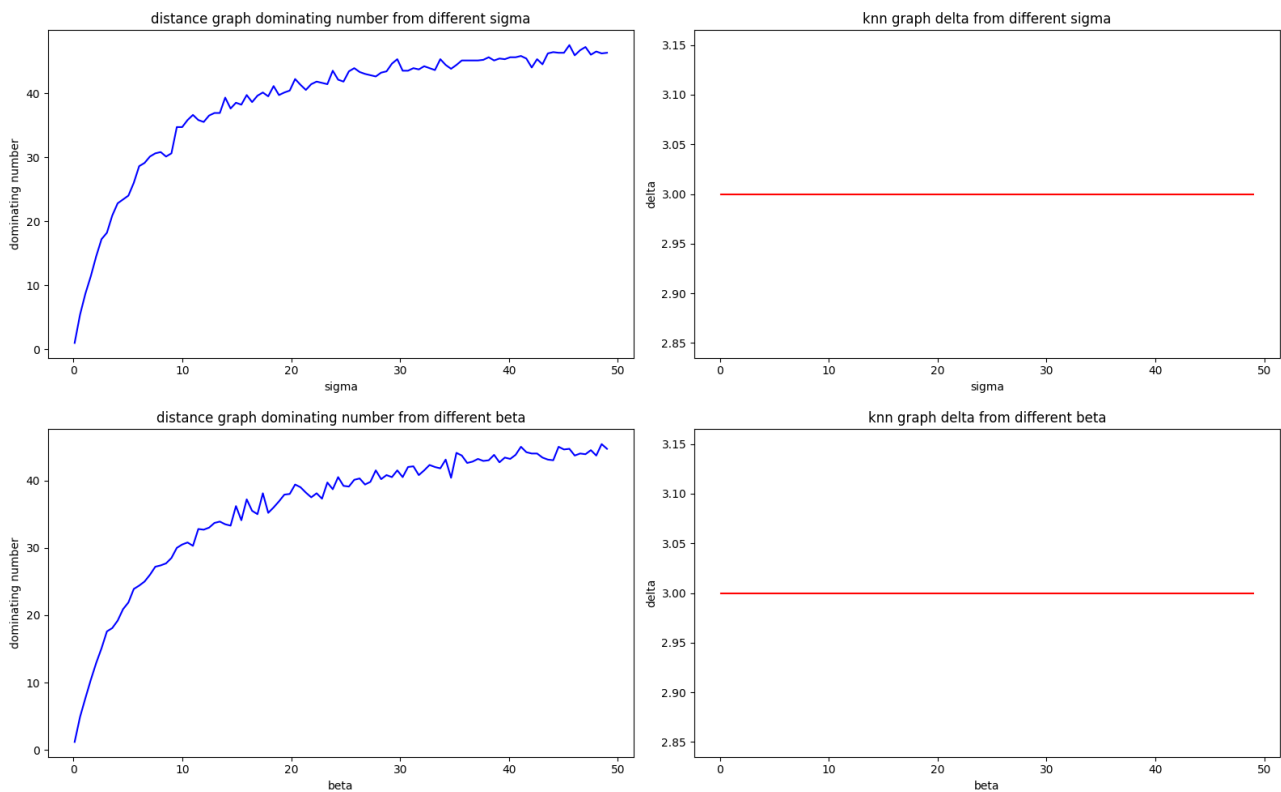
Из графиков заметно, что лишь с дистанционным графом хочется продолжать работать

### 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 экспериментов `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`. На двух графиках ниже показана зависимость среднего числа характеристик в зависимости от параметров распределений:



## Шаг 2. Фиксируем $\theta, v$ . Исследуем взаимосвязь между $n, k, d$ и $T^{KNN}, T^{dist}$

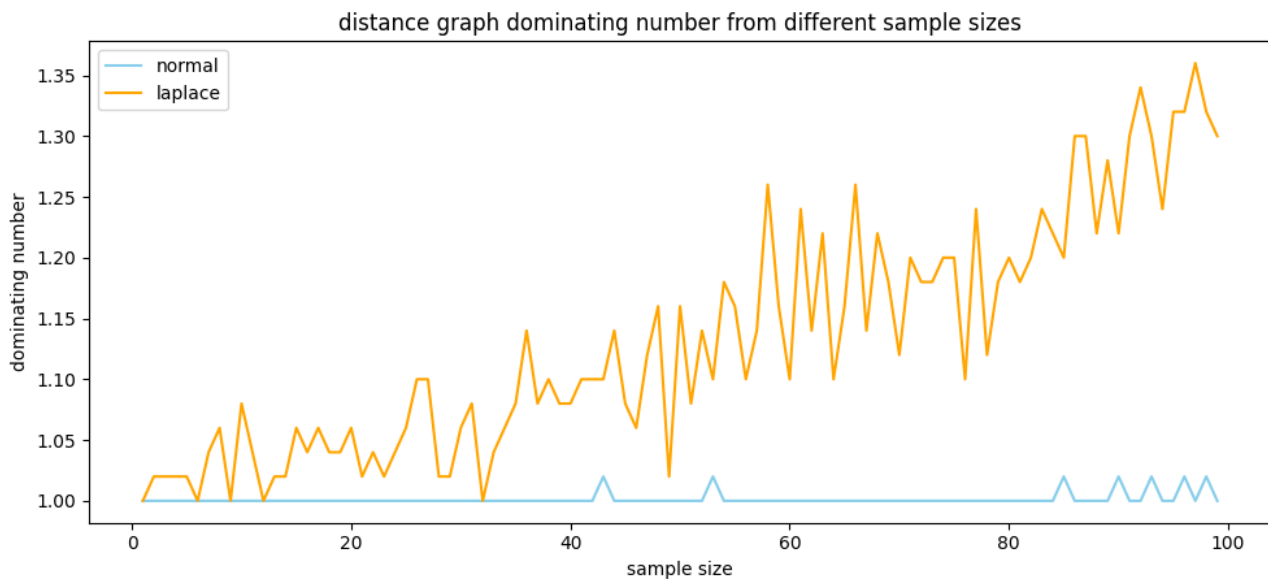
### Результаты Соколовского С.П.

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.

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  $d \geq 3$ , то характеристика графа из нормального распределения становится почти всегда равной 1, а вот при распределении Лапласа немного больше. Снизу график зависимости среднего числа доминирования от размера выборки при  $d = 3.5$ :

## Compare normal and laplace distributions

**Шаг 3. Фиксируем  $\theta, v$ . 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  $\mathcal{A}$ , которое должно удовлетворять двум условиям:

1. Контроль ошибки первого рода: вероятность ошибочно отвергнуть нулевую гипотезу  $H_0$  (данные имеют нормальное распределение) при её справедливости не превышает  $\alpha = 0.05$ .
2. Максимизация мощности: вероятность корректно отвергнуть  $H_0$  в пользу альтернативы  $H_1$  (например, распределение Лапласа) должна быть максимальной.

Множество  $\mathcal{A}$  конструируется итеративно следующим образом: На каждом шаге генерируется большое число выборок (`number_of_experiments`) из нормального распределения. Для каждой выборки вычисляется характеристика графа. Если значение характеристики не принадлежит текущему множеству  $\mathcal{A}$ , оно считается "ошибочным" (ложным отклонением  $H_0$ ). Ошибка первого рода оценивается как доля таких "ошибочных" случаев: Пока ошибка  $\text{err} > \alpha$ , в  $\mathcal{A}$  добавляется наиболее частое значение характеристики из "ошибочных" результатов (мода). Это снижает долю ошибок за счёт включения типичных для  $H_0$  значений. Процесс останавливается, когда  $\text{err} \leq \alpha$ , либо когда "ошибочные" значения исчерпаны.

## Часть II. Несколько характеристик проверки гипотезы

---

### Шаг 1. Исследуем важность характеристик

Результаты Соколовского С.П.

TODO

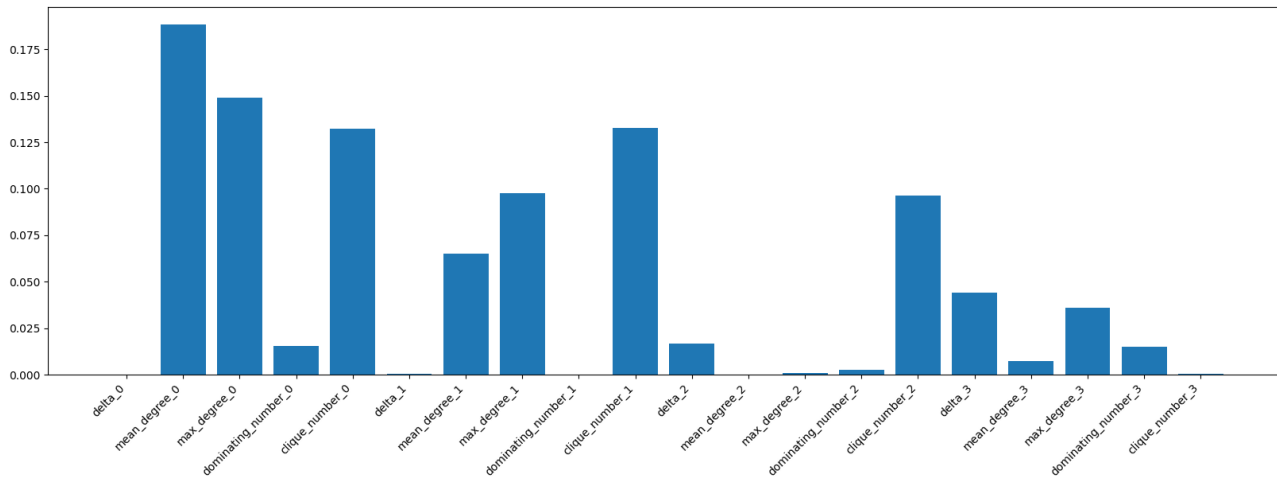
Результаты Григоренко М.Д.

В файле `experiments_second_part.ipynb` написан класс `DistributionClassifier`, принимающий на вход параметр  $n$  - размер выборки и модель классификации, которую предстоит обучить. Для выявления признаков по данной выборке строится 4 дистанционных графа с различным параметром  $d$ , для каждого графа считаются следующие характеристики:

1. Минимальная степень вершины
2. Средняя степень вершины
3. Максимальная степень вершины
4. Число доминирования
5. Кликовое число

Важность характеристик слабо меняется при разных  $n$ . Самыми важными характеристиками оказались средняя и максимальная степени дистанционного графа при  $d = 0.3$  (самое

маленькое значени  $d$ ). Ниже предствлена гистограмма важности признаков при  $n = 500$ :



## Шаг 2. Метрики качества для разных выборок

Результаты Соколовского С.П.

TODO

Результаты Григоренко М.Д.

Для каждого  $n$  (размер выборки) были обучены классификаторы `RandomForestClassifier`, с подобранными гиперпараметрами `max_depth` и `min_samples_split`, и `BaggingClassifier` с подобранным гиперпараметром `n_estimators` (из библиотеки `sklearn`). Для всех выборок чуть лучше (или примерно одинаково) оказался классификатор `RandomForestClassifier`. Лучшие метрики (за положительный класс взято  $H_1$  (распределение Лапласа)):

1.  $n = 25$ : accuracy = 0.77, precision = 0.79, recall = 0.74.. Измерение проводилось на 2000 примерах.
2.  $n = 100$ : accuracy = 0.95, precision = 0.94, recall = 0.96. Измерение проводилось на 200 примерах.
3.  $n = 500$ : accuracy = 1.0, precision = 1.0, recall = 1.0. Измерение проводилось на 200 примерах (из-за сложности построения дистанционного графа на 500 вершинах).

## Шаг 3. Выводы о вероятности ошибки первого рода и мощности подхода

Результаты Соколовского С.П.

TODO

Результаты Григоренко М.Д.

Мощность критерия по определению равна метрике `recall`. Вероятность ошибки первого рода была измерена отдельно:

1.  $n = 25$ : `error` = 0.22. Измерение проводилось на 2000 примерах.
2.  $n = 100$ : `error` = 0.055. Измерение проводилось на 2000 примерах.
3.  $n = 500$ : `error` = 0.0. Измерение проводилось на 200 примерах (из-за сложности построения дистанционного графа на 500 вершинах). Таким образом, можно оценить ошибку сверху `error` < 0.01.