Московский Государственный Университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Отчет по заданию №1 "**Метрические алгоритмы классификации**". Алгоритм k ближайших соседей

Кузьмин Н. В.
студент кафедры ММП
317 группа

Октябрь,
2019

Содержание

1 Введение

В этом документе представлен отчет о проделанных экспериментах по практическому заданию №1, анализ результатов.

2 Эксперименты

В этом блоке приведены все обязательные эксперименты, которые изложены в формулировке задания.

2.1 Скорость поиска ближайших соседей

2.1.1 Дизайн эксперимента:

Были протестированы 4 алгоритма поиска 5 ближайших соседей с разными размерами признакового пространства:

Алгоритмы:

- «my_own»
- «brute»
- «kd_tree»
- «ball_tree»

Размеры признакового пространства:

- 10
- 20
- 100

2.1.2 Ожидания

Ожидается, что «kd_tree», «ball_tree» будут очень хорошо работать для маленького количества признаков, но с увеличением размерности признакового пространства произойдет резкое увеличение скорости работы.

2.1.3 Результаты

Подробные результаты экспериментов приведены в таблице 1:

2.1.4 Выводы

Самым стабильным алгоритмом оказался «brute», который практически не деградирует с ростом размерности. Как и ожидалось, алгоритмы, реализованные на деревьях, сильно замедлились на признаковом пространстве размерности 100.

Таблица 1: Результаты эксперимента №1

размерность	алгоритм	время работы
10	my_own	68.07
	brute	7.84
	kd_tree	**0.44**
	ball_tree	1.72
20	my_own	76.82
	brute	7.95
	kd_tree	**1.40**
	ball_tree	6.63
100	my_own	78.31
	brute	**8.28**
	kd_tree	82.76
	ball_tree	97.67

2.2 Зависимость точности модели от количества соседей и метрики

2.2.1 Дизайн эксперимента

В этом эксперименте была рассмотрена зависимость точности и времени работы модели k ближайших от следующих параметров на 3 валидационных фолдах:

- k от 1 до 10 (только влияние на точность).
- Евклидова или косинусная метрики.

2.2.2 Результаты

Зависимость средней точности от числа соседей для различных метрик приведена на графике 1

Измерения скорости для евлидовой и косинусной метрик приведены в таблице 2

2.2.3 Выводы

Из графика 1 видно, что:

1. Наилучшая точность алгоритма достигается при $k = 3$ для обеих метрик.

Рис. 1:

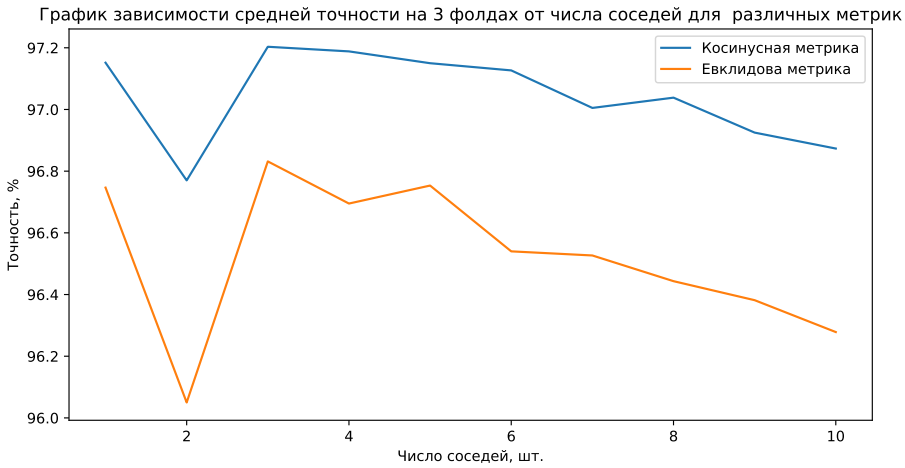


Таблица 2: Скорость работы метрик (измеренная при кросс-валидации на 3 фолдах)

метрика	время
евклидова	103.42
косинусная	101.69

2. С увеличением количества соседей (при $k \geq 5$) точность начинает убывать.

Из таблицы 2 видно, что скорость работы алгоритмов практически не зависит от выбора данных метрик

2.3 Сравнение точности взвешенного метода и метода без весов

2.3.1 Дизайн эксперимента

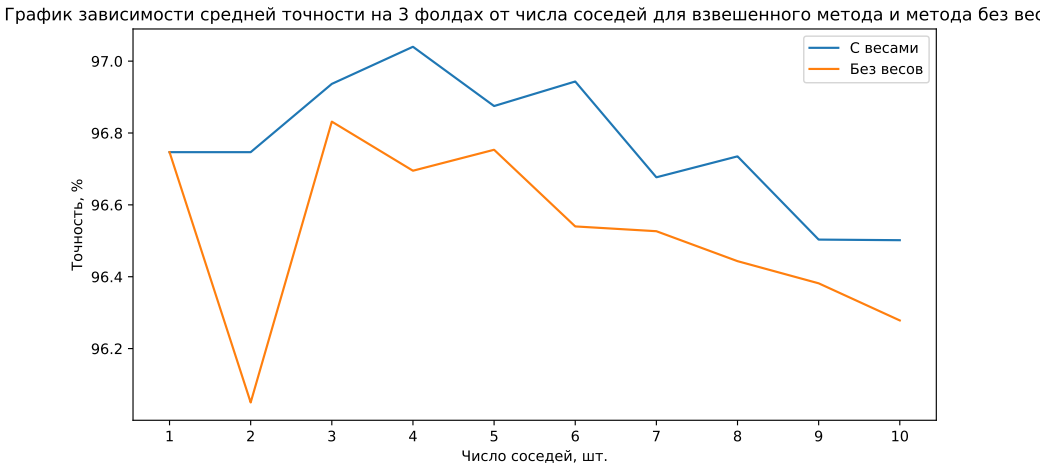
2.3.2 Результаты

Результаты эксперимента №3 приведены на графике 2

2.3.3 Выводы

По графику 2 можно заметить, что взвешенный метод лучше по точности во всех случаях, кроме $k = 1$, что является логичным, так как в этой

Рис. 2:



ситуации веса бесполезны.

2.4 Анализ модели с лучшим качеством

2.4.1 Дизайн эксперимента

Применим лучший алгоритм к исходной обучающей и тестовой выборке и проанализируем матрицу ошибок (confusion matrix).

2.4.2 Результаты

Матрица ошибок представлена на рисунке №, Визуализированные объекты - на рисунке №

2.4.3 Выводы

Можно увидеть, что самые частые ошибки происходят на парах цифр, представленных ниже:

- 4 и 9
- 1 и 7
- 3 и 5
- 3 и 8

Эти пары похожи по написанию, поэтому алгоритму тяжелее отличить их друг от друга.

Рассмотрим теперь объекты, на которых произошли ошибки. У них можно заметить характерные особенности:

1. Засечки
2. Крючки
3. Разные деформации

В большинстве случаев видно, что написаннные цифры немного деформированы и похожи на другие в некоторых чертах. Эти особенности затрудняют классификацию для алгоритма.