МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В.Ломоносова

Факультет вычислительной математики и кибернетики

Отчет по заданию №1

«Метрические алгоримы классификации»

Выполнила:
студентка 317 группы
Анисимова Д. В.

Москва
2021

Содержание

1 Введение

В задании необходимо было реализовать алгоритм K ближайших соседей, исследовать его качество и скорость в зависимости от различных параметров. Эксперименты были проведены на датасете MNIST.

2 Эксперименты

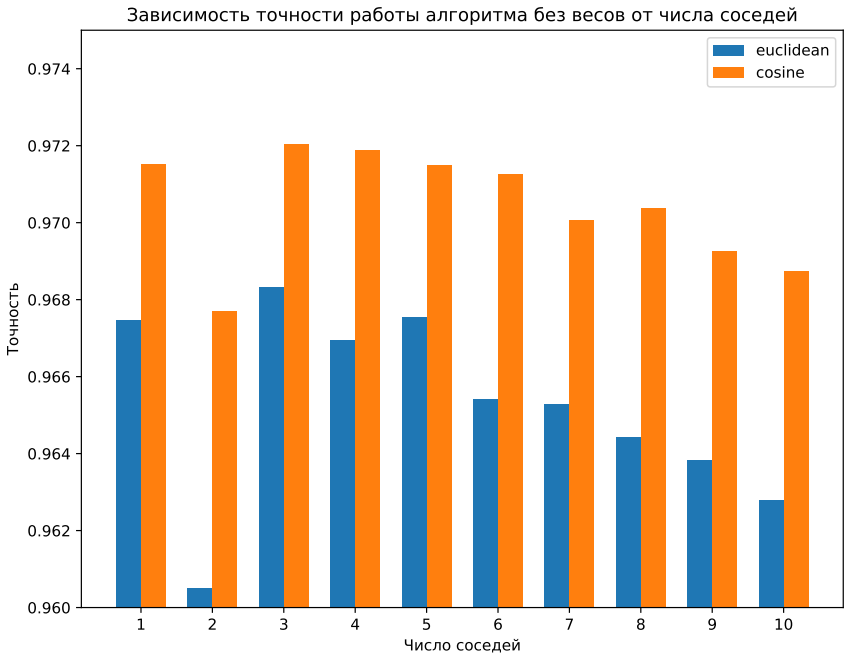
2.1 Сравнение алгоритмов поиска ближайших соседей

Число признаков	my_own	brute	kd_tree	ball_tree
10	75.16	14.64	3.51	5.61
20	76.14	14.77	2.55	34.15
100	80.05	19.65	171.18	168.02

Таблица 1: Время работы алгоритмов в зависимости от числа признаков

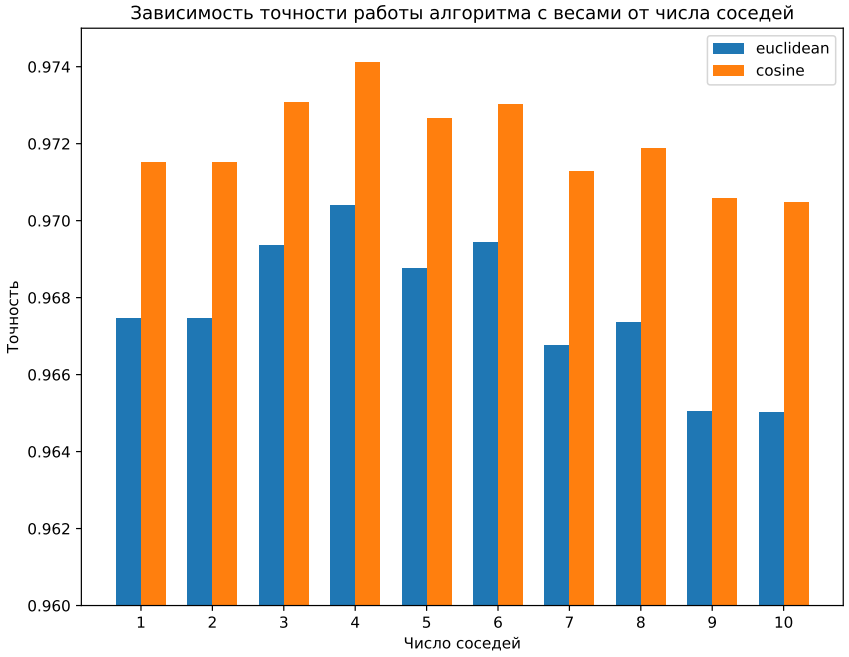
Из датасета было выбрано 10, 20 и 100 случайных признаков, для каждого из этих наборов было измерено время работы алгоритмов my_own, brute, kd_tree, ball_tree. Из результатов эксперимента видно, что при малом числе признаков быстрее работают алгоритмы kd_tree и ball_tree. При высокой размерности признакового пространства лучше всего работает brute, поэтому в дальнейших экспериментах будет использоваться этот метод, так как число признаков там достаточно велико.

2.2 Оценка точности и времени работы без учета весов

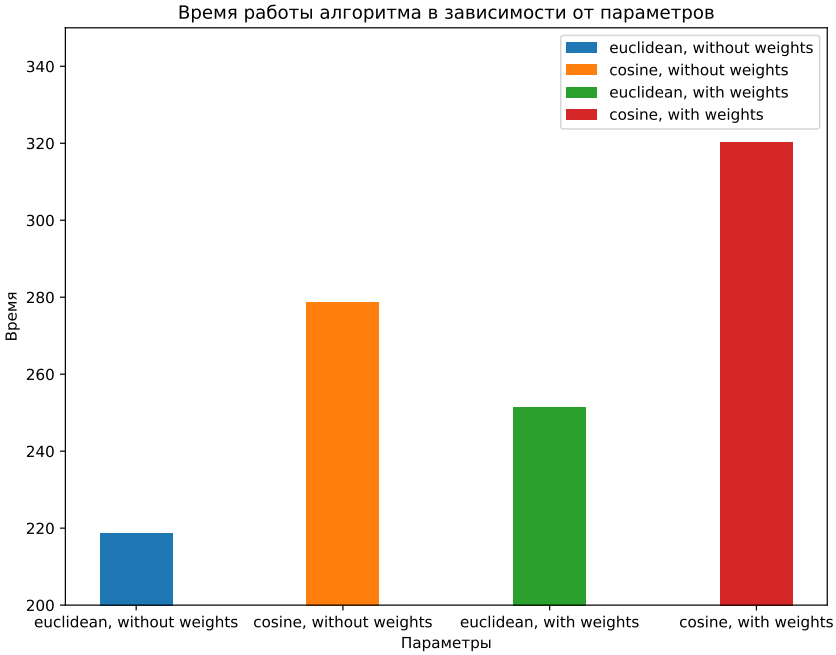


В этом эксперименте варьировалось 2 параметра: метрика и число соседей. В зависимости от этих параметров менялись время работы алгоритма и его точность. Как следует из графика, наибольшая точность достигается при косинусной метрике и числе соседей, равном 4. Однако при использовании косинусной метрики время работы алгоритма увеличивается.

2.3 Оценка точности и времени работы взвешенного метода



Эксперимент аналогичен эксперименту 2, но были учтены веса. Оказалось, что взвешенный метод работает точнее, чем метод без весов (то есть когда все веса принимаются равными 1). Наилучшая точность, как видно на графике, достигается при 4 соседях и косинусной метрике.
Поскольку на вычисление расстояний требовалось дополнительное время, взвешенный метод оказался немного медленнее:



В дальнейших экспериментах будет использоваться взвешенный метод с косинусной метрикой и 4 соседями для достижения наиболее высокой точности.

2.4 Анализ лучшего алгоритма

Тестовая выборка	Кросс-валидация	Лучший алгоритм (из интернета)
0.9752	0.9741	0.9979

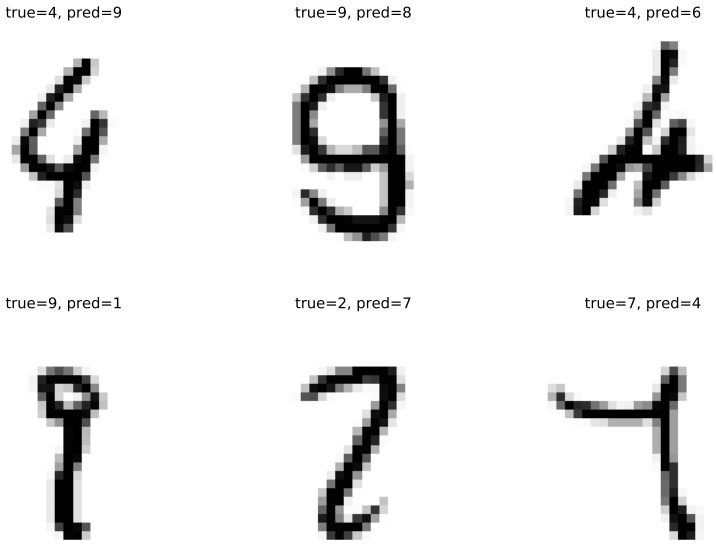
Таблица 2: Точность алгоритма на датасете MNIST

Из таблицы видно, что взвешенный метод 4 ближайших соседей на тестовой выборке дает почти такую же точность, как и на кросс-валидации, откуда следует, что переобучения не было. Однако эта точность ниже точности, указанной в интернете.

Цифра	0	1	2	3	4	5	6	7	8	9
0	977	1	0	0	0	0	1	1	0	0
1	0	1129	3	1	0	0	2	0	0	0
2	8	0	1009	1	1	0	0	8	5	0
3	0	1	3	976	1	12	0	4	9	4
4	2	1	0	0	946	0	6	2	0	25
5	4	0	0	9	1	863	7	1	4	3
6	3	3	0	0	1	3	948	0	0	0
7	2	10	4	0	1	0	0	998	0	13
8	7	1	2	9	3	3	5	4	936	4
9	7	7	2	5	7	3	1	4	3	970

Таблица 3: Матрица ошибок

Как видно из матрицы ошибок, чаще всего алгоритм неправильно классифицирует цифры 4 и 9, 7 и 9, 3 и 5, 7 и 1. Это происходит потому, что цифра написана не очень разборчиво, поэтому похожа на другую цифру:



2.5 Аугментация обучающей выборки

В результате поворотов, смещений и применений гауссовского фильтра обучающая выборка была увеличена в несколько раз. На кросс-валидации получились следующие результаты:

Параметр	Точность на кросс-валидации
Поворот, α=5	0.997
Поворот, α=10	0.974
Поворот, α=15	0.941
Смещение по x, shift=1	0.983
Смещение по x, shift=2	0.970
Смещение по x, shift=3	0.969
Смещение по y, shift=1	0.983
Смещение по y, shift=2	0.970
Смещение по y, shift=3	0.968
Фильтр, σ =0.5	1.0
Фильтр, σ =1	0.991
Фильтр, σ =1.5	0.980

Таблица 4: Точность на кросс-валидации после преобразования обучающей выборки

Отсюда видно, что при повороте на α=5 и фильтре σ =1 точность на кросс-валидации чрезвычайно высокая, поэтому, скорее всего, произошло переобучение. Смещения при shift=2, shift=3 и повороты при α=10, α=15 качество алгоритма не улучшают, поэтому не имеет смысла использовать эти преобразования ко всей обучающей выборке.
Исходя из этих рассуждений, применим к обучающей выборке сдвиг по x на 1 и фильтр с дисперсией 1. В результате качество улучшится, точность станет равной 0.9825. Разность исходной матрицы ошибок и матрицы ошибок после «размножения» обучающей выборки станет выглядеть так:

Цифра	0	1	2	3	4	5	6	7	8	9
0	-1	1	0	0	0	0	1	-1	0	0
1	0	-1	0	1	0	0	0	0	0	0
2	3	-1	-1	-1	0	0	-1	-3	4	0
3	0	1	2	-12	0	3	0	0	5	1
4	2	1	0	0	-15	0	2	0	0	10
5	2	-1	0	4	0	-9	3	0	2	1
6	0	0	0	0	0	0	0	0	0	0
7	2	1	-1	0	-2	-1	0	-4	0	5
8	4	1	0	7	1	-2	3	0	-15	1
9	7	6	3	2	3	1	1	-2	1	-15

Таблица 5: Разность матриц ошибок до аугментации тестовой выборки и после

Отсюда видно, что алгоритм стал лучше «отличать» цифры 4 и 9, 0 и 9, 3 и 8.

2.6 Аугментация тестовой выборки

В результате поворотов, смещений и применений гауссовского фильтра к тестовой выборке получились следующие результаты:

Параметр	Точность на тестовой выборке
Поворот, α=5	0.975
Поворот, α=10	0.976
Поворот, α=15	0.973
Смещение по x, shift=1	0.975
Смещение по x, shift=2	0.965
Смещение по x, shift=3	0.815
Смещение по y, shift=1	0.976
Смещение по y, shift=2	0.948
Смещение по y, shift=3	0.759
Фильтр, σ =0.5	0.975
Фильтр, σ =1	0.969
Фильтр, σ =1.5	0.964

Таблица 6: Точность работы алгоритма в результате аугментации тестовой выборки

Из этой таблицы видно, что качество в результате этих преобразований либо не меняется, либо ухудшается. Поэтому аугментация тестовой выборки смысла не имеет.
Рассмотрим матрицу ошибок в случае применения фильтра Гаусса с дисперсией 1 к тестовой выборке.

Цифра	0	1	2	3	4	5	6	7	8	9
0	976	1	0	0	0	0	2	1	0	0
1	0	1130	2	1	0	0	2	0	0	0
2	10	0	1002	1	1	0	2	9	7	0
3	3	3	3	968	0	11	0	5	12	5
4	2	1	2	0	926	0	8	3	2	38
5	11	1	0	10	1	850	6	1	8	4
6	5	2	0	0	1	1	949	0	0	0
7	2	15	4	0	2	0	0	987	1	17
8	8	4	2	5	2	4	4	4	938	3
9	6	10	2	4	5	3	1	6	7	965

Таблица 7: Матрица ошибок после аугментации тестовой выборки

Если сравнить эту матрицу с таблицей 3, станет видно, что алгоритм стал чаще «путать» цифры 9 и 0, 4 и 9, 3 и 8, то есть число ошибок увеличилось. Такие же результаты получим, применив остальные виды преобразований к тестовой выборке.

3 Вывод

В результате проведенных экспериментов было получено, что алгоритм 4 ближайших соседей показывает довольно высокую точность на датасете MNIST. Однако по сведениям из интернета он является не самым оптимальным.
Качество алгоритма удалось улучшить, используя комбинацию преобразований изоб-

ражений датасета. Однако при аугментации обучающей выборки алгоритм стал работать существенно медленнее.

4 Список литературы

1. https://www.kaggle.com/cdeotte/mnist-perfect-100-using-knn