Московский государственный университет имени М. В. Ломоносова
Факультет вычислительной математики и кибернетики

Кафедра ММП, компьютерный практикум

Отчет о выполненном задании «Метрические алгоритмы классификации»

Травникова Арина Сергеевна, 317 группа

Введение.

Задание заключается в реализации метрического алгоритма классификации методом ближайших соседей и исследовании его работы на датасете изображений рукописных цифр MNIST. В процессе работы проведены различные эксперименты для сравнения эффективности алгоритмов с разными параметрами и подбора лучших гиперпараметров по кросс-валидации. Также выполнено предсказание для тестовой выборки.

Постановка задачи и описание алгоритма.

Пусть дана обучающая выборка с известными ответами $\{(x_i, y_i)\}_{i=1}^N$, где $x_i \in \mathbb{R}^d$, $y_i \in \mathbb{Y} = \{1, \ldots, C\}$, и выборка объектов $\{z_i\}_{i=1}^M$, $z_i \in \mathbb{R}^d$, для которых необходимо предсказать ответ.

Метрический метод классификации заключается в том, что на множестве объектов вводится функция расстояния $\rho(z, x)$, и для произвольного объекта z производится упорядочение обучающей выборки по расстоянию до z:

$$p(z, x_{(1)}) \leq p(z, x_{(2)}) \leq \ldots \leq p(z, x_{(N)})$$

Предсказание класса для z основывается на ответах для k его ближайших соседей:

$$a(z) = \arg\max_{y \in Y} \sum_{i=1}^{k} \big[x_{(i)} = y\big] w_i$$

Здесь $w_i = f(i, \rho(z, x_{(i)}))$ - функция весов.

Гиперпараметрами модели являются функция расстояния $\rho(z, x)$, число соседей k и веса w_i, которые подбираются по кросс-валидации.

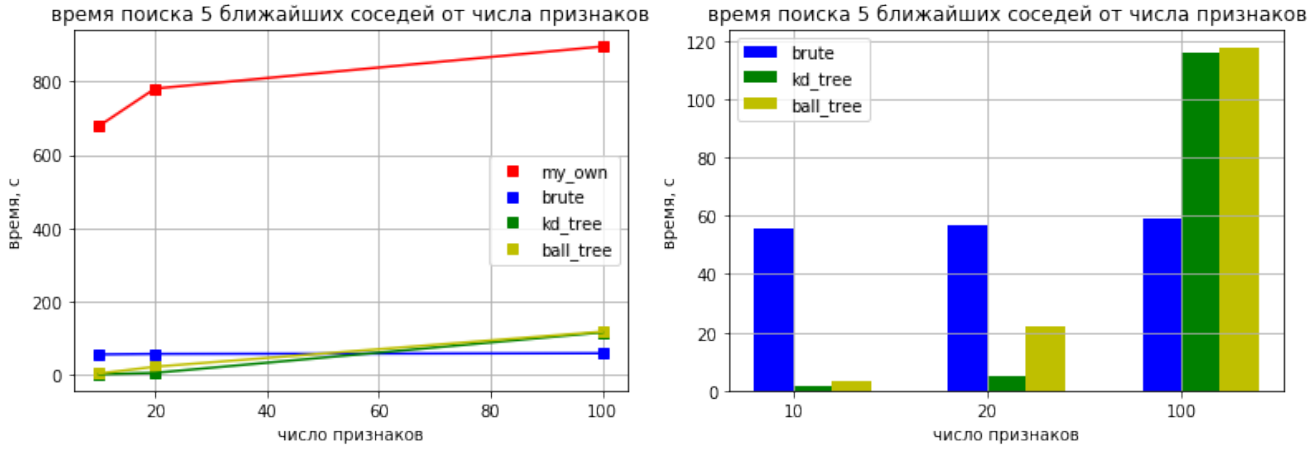
Датасет MNIST состоит из 70000 изображений цифр от 0 до 9 размером 28×28 пикселей, первые 60000 объектов относятся к обучающий выборке, последние 10000 - к тестовой.

Эксперимент 1.

Эксперимент состоит в исследовании зависимости времени работы функции поиска 5 ближайших соседей в евклидовой метрике от алгоритма поиска: *«brute»*, *«kd_tree»*, *«ball_tree»* , *«my_own»*.

При проведении работы первые 3 реализации взяты с sklearn.neighbors.NearestNeighbors, последняя - собственная реализация, которая заключается подсчете евклидова расстояния между объектами тестовой и обучающей выборки в векторизованной форме, сортировке расстояний и выборе k ближайших точек для каждого объекта теста.

Измерение времени работы производится на объектах тестовой выборки датасета MNIST с различным числом случайно выбранных признаков: $10, 20, 100$ признаков. Результаты представлены на рис. 1



a) b)

Рис. 1: Зависимость времени работы алгоритма поиска ближайших соседей от количества признаков.

Результаты показывают, что при малом количестве признаков наиболее быстрыми являются алгоритмы *Ball tree* и *K-d tree*, так как они выполняют структурирование признакового пространства для более эффективного поиска ближайших соседей. При небольшом количество признаков сложность поиска для *K-d tree* логарифмически зависит от числа объектов обучающей выборки, а сложность полного перебора - линейно, поэтому в пространстве признаков низкой размерности целесообразно использование *K-d tree* алгоритма.

Однако с ростом размерности использование деревьев теряет смысл: чем больше число признаков, тем ниже эффективность структурировани пространства из-за того, что все объекты становятся одинаково далеки друг от друга и группы ближайших перестают быть геометрически компактными. Сложность поиска для деревьев обретает линейный порядок, но скорость роста функции будет выше, чем у переборного алгоритма, из-за дополнительных затрат на организацию структур данных.

С учетом того, что размерность пространства признаков для MNIST равна 784, далее

для проведедения экспериментов будем использовать *«brute»* подход для поиска ближайших соседей в целях экономии времени.

Эксперименты 2, 3.

Эксперимент состоит в исследовании зависимости времени работы алгоритма k ближайших соседей и точности, оцененной по кросс-валидации, от настраеваемых гиперпараметров:

- k от 1 до 10 - число ближайших соседей
- использование евклидовой или косинусной метрики

$$\rho_{euc}(x,y) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2},\ \rho_{cos}(x,y) = 1 - \frac{\langle x,y\rangle}{\|x\|\|y\|}$$

- использование взвешенного и невзвешенного учета наблюдений

  При работе с весами предсказание для объекта z определяется формулой

$$a(z) = \arg\max_{y\in Y}\sum_{i=1}^{k}\big[x_{(i)} = y\big]w_i,\ w_i = \frac{1}{\rho(z,x_{(i)}) + \varepsilon},\ \varepsilon = 10^{-5}$$

  Для удобства реализации в невзвешенном методе используется та же формула, но $w_i = 1$

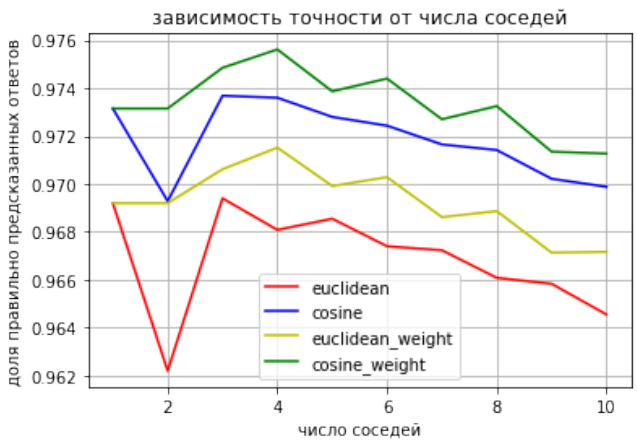


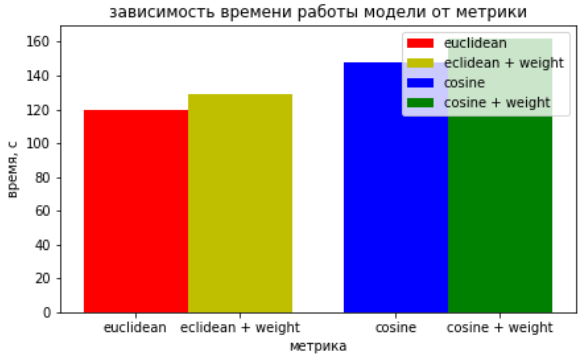
Рис. 2: Зависимость доли правильно предсказанных ответов от числа ближайших соседей.

Под точностью алгоритма будем понимать долю правильно предсказанных ответов. Измерение точности работы моделей осуществляется на всей обучающей выборке по кросс-валидации с 3 фолдами.

Чтобы вычислить точность, для каждого числа ближайших соседей произведено усреднение точности, полученной при кросс-валидации на всех валидационных блоках. Также для более корректной оценки предварительно выполнено перемешивание обучающей выборки, чтобы все элементы одного класса не попадали в один фолд.

Результаты показывают, что евклидова метрика проигрывает в точности косинусной для любого числа ближайших соседей (разница порядка 0.4%), а невзвешенный учет наблюдений - взвешенному. Наибольшая разница между алгоритмом с весами и без достигается при $k = 2$: если 2 ближайших соседа будут иметь разный класс, то во взвешенном методе выберется именно класс ближайшего объекта из-за большего веса, в в невзвешенном - произвольный. Скользящий контроль указывает оптимальные значения параметра k: $k = 4$ для метода с весами и $k = 3$ иначе.

Измерение времени работы тоже производится на обучающей выборке по кросс-валидации с 3 фолдами: засекается общее время для предсказания ответов на всех валидационных блоках при количестве соседей $k = 10$. Результаты представлены на рис. 3



метрика	**время работы, с**
euclidean	119.2
euclidean_weight	129.0
cosine	148.2
cosine_weight	161.9

Рис. 3: Зависимость времени работы алгоритма k ближайших соседей от метрики и учета весов.

В результате эксперимента выяснили, что алгоритм с использованием косинусной меры работает существенно дольше, чем с евклидовой. Это объясняется тем, что вычисление расстояния по косинусу требует большего числа операций и работает медленнее. Следует отметить, что добавление весов не сильно влияет на время работы алгоритма в силу одинаковой реализации функции предсказания, кроме векторизованного вычисления w_i, которое вносит незначительный вклад в общее время работы.

Таким образом, при выборе метрики стоит учитывать требования к итоговому алгоритму: с косинусной мерой модель будет более точной, но менее эффективной по времени.

Эксперимент 4.

Эксперимент состоит в применении алгоритма с лучшими гиперпараметрами, подобранными по кросс-валидации, к тестовой выборке и анализе результатов. При проведении предыдущих исследований было выяснено, что оптимальной является модель со следующими параметрами:

- $k = 4$ - число ближайших соседей

- используется косинусная метрика
- используется взвешенный учет наблюдений ($w_i = \frac{1}{\rho(z,x_{(i)})+\varepsilon}$)
- используется алгоритм поиска ближайших соседей методом полного перебора - *«brute»*

Точность определяется как доля правильно предсказанных ответов. При указанных параметрах точность равняется 97.52%, в то время как лучшая известная точность классификации цифр для MNIST достигается методом *Regularization of Neural Networks using DropConnect* и равна 99.79%

Посмотрим, как соотносится точность, оцененная по кросс-валидации, и точность на тесте для различного количества ближайших соседей.

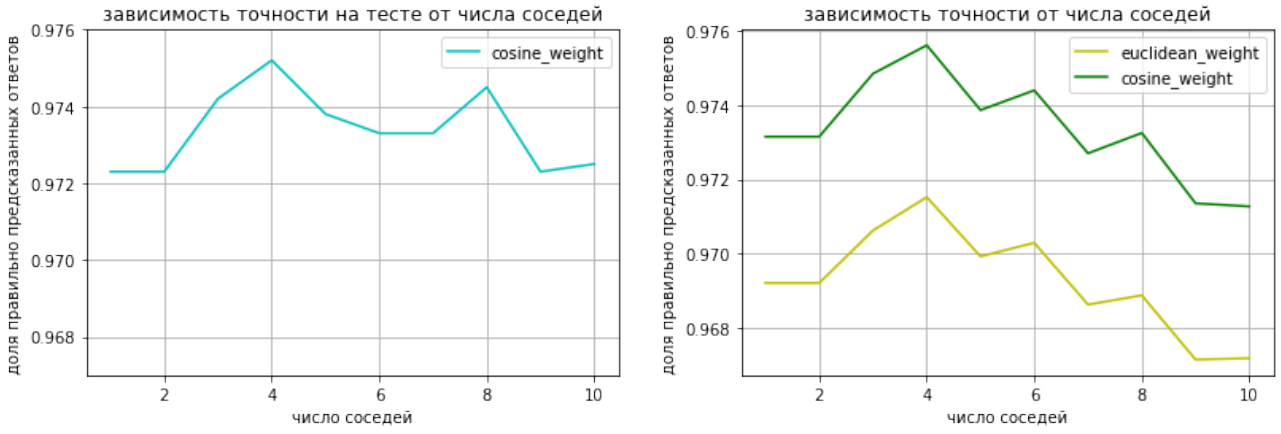


Рис. 4: Зависимость точности предсказания алгоритма от числа соседей для тестовой выборки и кросс-валидации

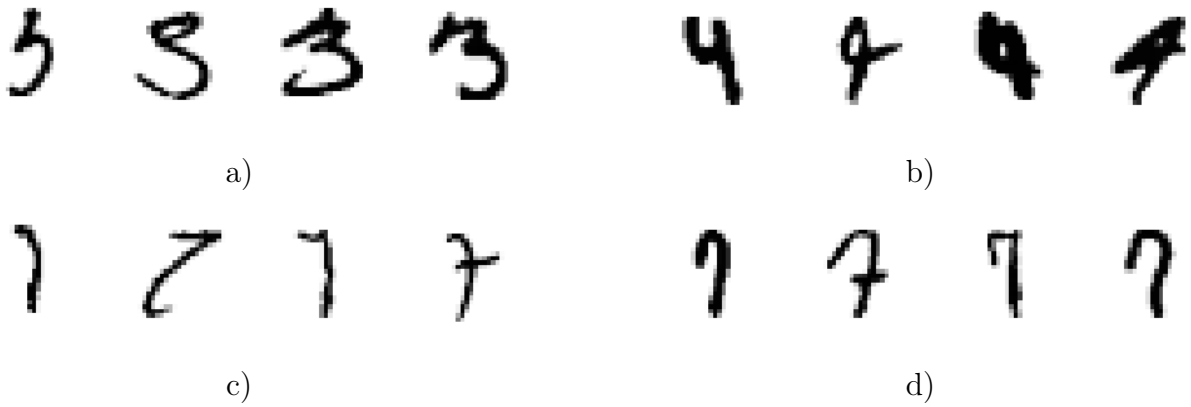
График показывает, что зависимость точности от параметра k по кросс-валидации на обучающей выборке хорошо приближает поведение функции точности и для тестовой выборки, что подтверждает корректность настройки параметров методом скользящего контроля.

Для повышения качества предсказания алгортма следует проанализировать матрицу ошибок, представленную ниже в таблице 1. Элемент в позиции (i, j) означает количество цифр i, распознанных как j. То есть диагональные элементы показывают количество верно предсказанных ответов, а внедиагональные - ошибок.

Обращаясь к таблице 1, замечаем, что наиболее часто ошибки происходят при распознавании цифры 4 (принимается за 9), 7 (принимается за 9 и 1), 3 (принимается за 5). Общее число ошибок - 248.

	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 1: Матрица ошибок предсказания для тестовой выборки.



a) b)

c) d)

Рис. 5: Изображения, дающие ошибку распознавания: a) 3 принимается за 5, b) 4 принимается за 9, c) 7 принимается за 1 , d)7 принимается за 9

На рис. 5 показаны изображения, на которых алгоритм дает ошибочное предсказание класса. Приведенные рукописные цифры обладают некоторыми общими чертами, как-то: цифра не дописана до конца или сливаются некоторые линии, увеличен наклон в одну из сторон, размытие. Можно предположить, что поворот изображений обучающей выборки и размытие по Гауссу помогут исправить некоторые ошибки.

Эксперимент 5.

Эксперимент заключается в увеличении объема обучающей выборки путем применения некоторых элементарных преобразований к изображениям, оставляющих их внутри своего класса; исследовании точности метода ближайших соседей с размноженной обучающей выборкой и применении лучших преобразований для предсказания на тесте.

В ходе работы рассматриваются следующие преобразования изображений:

- поворот, величина угла: $5, 10, 15$ градусов

- смещение, величина сдвига: $1, 2, 3$ пикселя
- фильтр Гаусса, величина дисперсии: $0.5, 1, 1.5$

Рассмотрим первый подход к выбору преобразований, основанный на подборе лучшего параметра для каждого элементарного преобразования по кросс-валидации. Для определения лучшего преобразования произведено увеличение выборки путем применения одного преобразования с заданным параметром и вычислена точность по кросс-валидации с 3 фолдами. Результаты эксперимента в таблице 2.

При размножении выборки с помощью каждого преобразования точность по кросс-валидации повысилась, лучшими преобразованиями стали поворот на 10 градусов в обе стороны, смещение на 1 пиксель по каждой размерности, фильтр Гаусса с дисперсией 1.

Без преобразований, 60000 объектов	точность,%	97.5		
Смещение, 300000 объектов	сдвиг	1	2	3
	точность, %	**98.3**	97.8	97.6
Фильтр Гаусса, 120000 объектов	дисперсия	0.5	1	1.5
	точность, %	97.8	**98.2**	98.1
Поворот, 180000 объектов	угол	5	10	15
	точность, %	98.1	**98.3**	98.1

Таблица 2: Средняя точность алгоритма по кросс-валидации для обучающей выборки, размноженной различными преобразованиями.

Применим метод ближайших соседей с обучающей выборкой, размноженной выбранными преобразованиями по отдельности и выборкой, полученной применением всех преобразований объемом 480000 объектов.
Также попробуем применить «случайный» подход к выбору преобразований: увеличим выборку до 240000 объектов, выбирая для каждого изображения преобразование и параметр случайным образом.

Преобразование	Точность, %
Без преобразований, 60000 объектов	97.52
Смещение на 1 пиксель, 300000 объектов	97.98
Фильтр Гаусса с дисп. 1, 120000 объектов	**98.13**
Поворот на 10 градусов, 180000 объектов	**98.13**
Случайный выбор, 240000 объектов	97.87
Все преобразования, 480000 объектов	**98.50**

Таблица 3: Точность алгоритма на тесте при размноженной обучающей выборке

Тестирование (таблица 3) показало, что лучшими однократными преобразованиями являются поворот на 10 градусов и размытие Гаусса с дисперсией 1: они дают повышение точности на 0.5%. Увеличение выборки в 8 раз позволяет повысить долю правильно предсказанных объектов на 1%. Случайное преобразование тоже дает улучшение качества, но незначительное.

Посмотрим, как изменилась матрица ошибок для моделей, обученных на выборках, которые размножены лучшим образом: применением поворота на 10 градусов и объединением всех однократно преобразованных выборок.

	0	1	2	3	4	5	6	7	8	9
0	976	1	0	0	0	0	2	1	0	0
1	0	1132	2	0	0	0	0	1	0	0
2	6	2	1008	2	1	0	1	**10**	2	0
3	0	0	1	986	1	7	0	3	7	5
4	1	0	0	0	952	0	5	2	0	**22**
5	3	0	0	6	1	874	3	2	1	2
6	2	3	0	0	0	1	951	0	1	0
7	2	7	3	0	0	0	0	1008	0	8
8	2	0	3	5	4	3	2	3	947	5
9	4	4	1	5	5	2	1	5	3	979

Таблица 4: Матрица ошибок предсказания для тестовой выборки при обучающей выборке, размноженной применением поворота на 10 градусов.

	0	1	2	3	4	5	6	7	8	9
0	976	0	0	0	0	0	3	1	0	0
1	0	1132	3	0	0	0	0	0	0	0
2	5	1	1013	0	1	0	2	**10**	0	0
3	0	0	2	992	1	4	0	3	4	4
4	0	0	0	0	962	0	4	2	0	**14**
5	2	0	0	5	1	875	6	1	0	2
6	2	2	0	0	0	1	953	0	0	0
7	1	7	4	0	0	0	0	1009	0	7
8	2	0	3	2	2	4	3	4	952	2
9	1	3	1	3	5	2	1	4	3	986

Таблица 5: Матрица ошибок предсказания для тестовой выборки при обучающей выборке, состоящей из объединения всех однократно преобразованных выборок.

Сравнение таблицы 4 и таблицы 1 показывает, что для всех цифр, кроме 2, количество правильных ответов классификатора увеличилось, что объясняется наличием ошибок, возникших из-за необычного наклона цифр в тестовой выборке.

Обратим внимание на характер ошибок при распознавании цифры 2. Из матрицы ошибок видно, что появилось много неверных ответов с предсказанием 2 как 7 (рис. 6a)

a) b)

Рис. 6: a) Изображения, дающие ошибку распознавания - 2 принимается за 7. b) Изображения цифры 7, повернутые на 10 градусов

.

Приведенные изображения помогают найти возможную причину ошибки: при повороте против часовой стрелки цифра 7 становится похожей на недописанную цифру 2, на изображениях которой и происходят неправильные предсказания.

При классификации тестовых объектов алгоритмом, использующим лучшую размноженную выборку, количество ошибок уменьшается с 248 до 150. Сравнение таблицы 5 с таблицей 1 показывает, что многие внедиагональные значения в матрице ошибок уменьшились, кроме строки 2, где ушли старые ошибки, но появились новые.

Таким образом, рамножение обучающей выборки применением различных преобразований к изображениям, оставляющих их в своем классе, позволяют повысить точность предсказания алгоритма в разной степени.

Эксперимент 6.

Эксперимент состоит в увеличении объема тестовой выборки путем применения некоторых элементарных преобразований к объектам, оставляющих их в своем классе, и изучении влияния такого размножения выборки на точность предсказания классификатора.

Размножение тестовой выборки происходит следующим образом: для каждого объекта z тестовой выборки построим множество объектов $\Phi(z) = \{\phi_i(z)\}_{i=0}^{p}$, где $\{\phi_i(z)\}$ - некоторое элементарное преобразование, $\phi_0(z)$ - тождественное преобразование.

Чтобы определить класс z, найдем k ближайших соседей из обучающей выборки для всех элементов множества $\Phi(z) = \{\phi_i(z)\}_{i=0}^{p}$ и выберем среди них те k, до которых рас-

стояние минимально: $x_{(1)} \dots x_{(k)}$. Тогда класс z определяется формулой:

$$a(z) = \arg\max_{y \in Y} \sum_{i=1}^{k} \big[x_{(i)} = y\big] w_i$$

.

Применим метод ближайших соседей для тестовой выборки, размноженной с лучшими выбранными параметрами каждого преобразования (аналогично эксперименту 5).

По результатам измерения точности на тесте оптимальным способом размножения тестовой выборки являются однократные преобразования - смещение на 1 пиксель и поворот на 10 градусов, а также объединение полученных множеств.

Преобразование	Точность, %
Без преобразований, 10000 объектов	97.52
Смещение на 1 пиксель, 50000 объектов	**97.98**
Фильтр Гаусса с дисп. 0.5, 20000 объектов	97.29
Поворот на 10 градусов, 30000 объектов	**98.02**
Все преобразования, 800000 объектов	97.98
Поворот и смещение, 70000 объектов	**98.21**

Таблица 6: Точность алгоритма на тесте при размноженной тестовой выборке

При анализе полученных результатов можно заметить, что размножение тестовой выборки размытием Гаусса только понижает качество работы алгоритма даже с минимальным значением дисперсии, а остальные преобразования дают меньшее приращение точности, чем при увеличении обучающей выборки.

Падение точности при размножении выборки преобразованием фильтр Гаусса можно объяснить тем, что выборка увеличивается всего 2 раза: если для первоначального изображения найдены верные близкие объекты, а для измененного ближайшие соседи найдены в другом классе, то может произойти так, что они окажутся ближе первых и тогда ответ будет неверный. Это показывает матрица ошибок для выборки, размноженной размытием Гаусса с дисперсией 1 (таблица 7): при сравнении с матрицей ошибок предсказания без преобразований выборки (таблица 1) можно заметить, что не произошло исправления каких-либо ошибок, но появились 120 новых. Особенно выросло число ошибок в распознавании 4. Как было показано на рис. 5b), размытая цифра 4 действительно похожа на 9, поэтому в этом случае фильтр Гаусса относит к ней еще больше изображений.

Таким образом, ближайшие обекты, найденные для измененного изображения тестовой выборки, могут не быть близкими для начального, а при увеличении обучающей выборки гарантируется близость именно к объекту теста, что делает этот метод точнее.

	0	1	2	3	4	5	6	7	8	9
0	976	1	0	0	0	0	1	1	1	0
1	0	1130	2	2	0	0	1	0	0	0
2	12	0	998	2	0	0	2	9	9	0
3	2	2	4	971	0	7	0	5	13	6
4	4	3	2	0	908	0	**11**	2	2	**50**
5	10	2	0	17	1	832	14	0	10	6
6	7	4	0	0	2	1	943	0	1	0
7	4	17	9	0	1	0	0	980	1	16
8	7	9	3	4	2	3	4	7	933	2
9	10	10	1	3	4	2	1	7	10	961

Таблица 7: Матрица ошибок предсказания для тестовой выборки, увеличенной размытием Гаусса с дисперсией 1. Число ошибок - 368.

Вывод.

Проведенные эксперименты показали, что метрический алгоритм классификации с использованием некоторых эвристик дает хорошую точность на датасете изображений цифр MNIST. Однако основной проблемой метода ближайших соседей является низкая скорость работы: при большой размерности пространства не существует алгоритма поиска ближайших объектов с асимптотикой лучше, чем $O(ND)$, где N - число объектов обучающей выборки, D - число признаков.

В общем случае для качественной работы алгоритма ближайших соседей важно подобрать оптимальные параметры под конкретную задачу: метрику, функцию весов, число соседей. Скользящий контроль позволяет корректно это сделать.