Задание 1. Метрические алгоритмы классификации

Кулик Андрей

Практикум 317 группы
15 октября 2020

1 Введение

В данной работе был реализован метрический алгоритм классификации методом ближайших соседей, и исследована его работа на реальных данных. Были проведены различные эксперементы для сравнения эффективности алгоритмов и подбора лучших гиперпарамметров.

2 Исследуемые данные

Для проведения эксперементов был загружен датасет MNIST рукописных чисел. Этот набор состоит из 70 000 изображений, каждой размером 784 пиксела, то есть 28 x 28 пикселов. При этом база данных делиться на обучающую выборку(60 тыс объектов) и тестовую (10 тыс объектов).

3 Эксперементы

3.1 Время работы стратегий в зависимости от количества признаков

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

Первые 3 реалиции были сымпортированы из библиотеки skrearn. Последняя - реализована самостоятельно, заключается в подсчете евклидова расстояния между объектами тестовой и обучающей выборки, сортировки расстояния и выборе k-ближайших соседей для каждого объекта.

Для выборки выбираем подмножество признаков размера 10, 20, 100 и тестируем на наших стратегиях, замеряя время работы программы(Рис.1)

Можно заметить, что методы *«brute»*, *«kd_tree»*, основанные на построеннии деревьев, показывают хорошие результаты только на пространствах малых размерностей. Это обусловлено тем, что они выполняют структурирование признакового пространства, что уменьшает эффективность на большом количестве признаков, так как группы ближайших соседей перестают быть компактными. Алгоритмы *«ball_tree»*, *«my_own»* наоборот показывают хорошие результаты на больших размерностях. Поэтому в дальнейших экспериментах я буду использовать именно эти реализации.

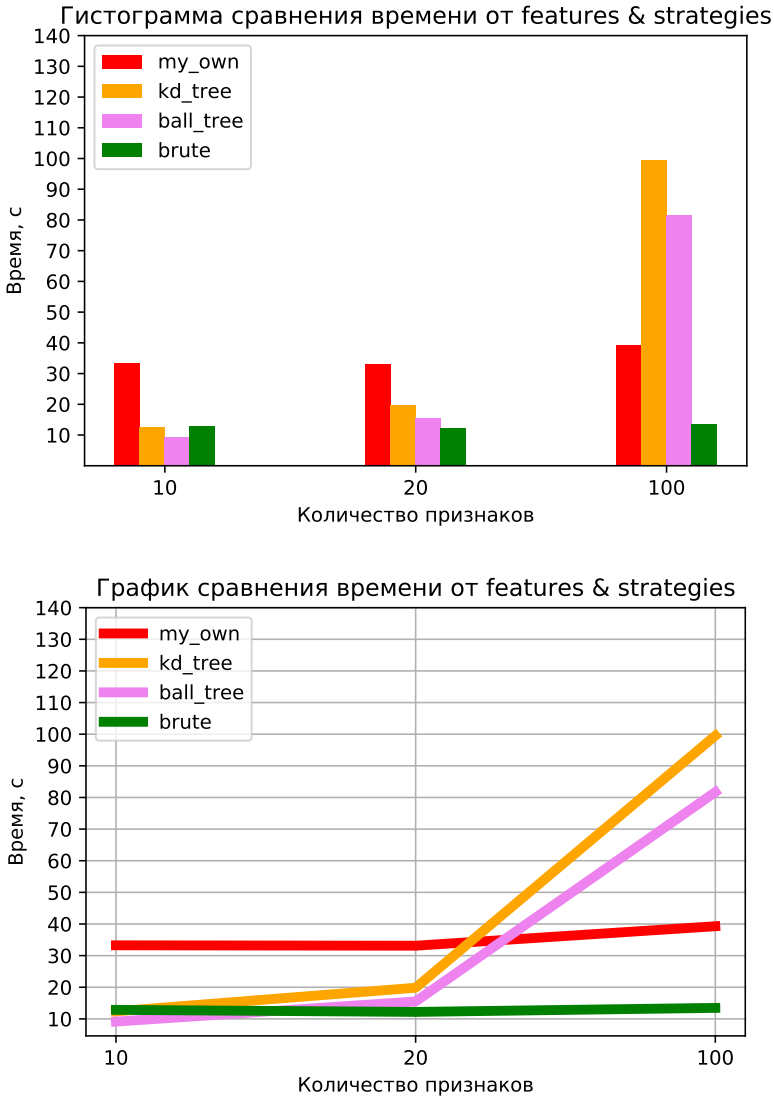


Рис.1 Время работы алгоритмов при разном количестве признаков(гистограмма и график)

3.2 Сравнение точности и времени работы метода с различными параметрами по кросс-валидации

Эксперемент заключается в оценки по кросс-валидации с 3 фолдами точность и время работы k ближайших соседей в зависимости от следующих факторов:

1. k от 1 до 10 (только влияние на точность).

2. Используется евклидова или косинусная метрика.

На Рис.2 представлена зависимость точности от количества соседей. Была мспользована *«brute»* стратегия, так как она показала наилучший результат по скорости работы. Расстояние были найдены без учитывания весов.Далее в эксперементах будет использовать точность - 'accuracy'(доля правильно предсказанных ответов). На графике видно, что точность пр учитывании 2 соседей снижается. Да и в целом, при учитвании нечетного количества невзвешенных соседей точность выше. Это связано со спецификой метода *np.argmax()*, которые при равных значениях элементов возвращает наименьший индекс, что не всегда верно.

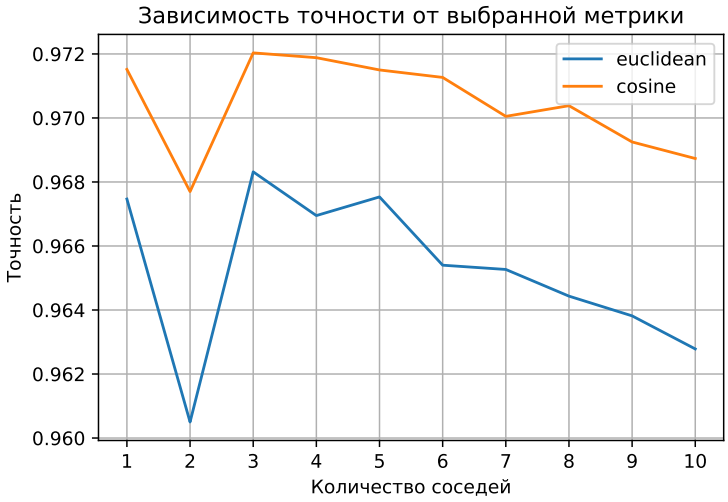


Рис.2 Зависимость точности от количества соседей и метрики

На графике выше видно, что косинусная метрика показывает результаты лучше, чем евклидова. Это связно с тем, что при подсчете евклидового расстояния большую роль играет насыщенность каждого пикселя, то есть числовое значение признака. Следовательно, цифры, совпадающие по форме, но отличные по насыщенности цвета(на это влияет,например, цвет ручки или нажим) в евклидовом расстоянии будут находиться дальше друг от друга, чем в косинусном, что может привести к неверному прогнозированию. Однако работает косинусная метрика дольше, на подсчет всех расстоний при 3 фолдах в общей сумме было потрачено **198.2с**, когда у евклидовой метрики это заняло **172.3 с**. Наилучший показатель по точности является **0.972** при 4 ближайших соседях.

3.3 Сравнение взвешенного метода с методом без весов

Эксперемент был проведен для косинусного расстояния с 3 фолдами кросс-валидации(Рис. 3). Видно, что точность, при учитывании весов для любых параметров k, выше, чем без учета весов. Особенно заметным является показатель при $k = 2$, за счет весов показатель точности не проседает.

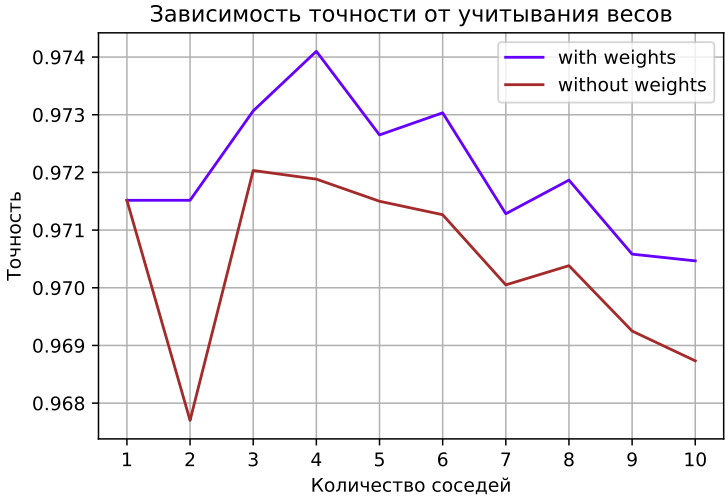


Рис.3 Зависимость точности при учитывании весов

3.4 Анализ работы алгоритма с лучшими параметрами

По результатам предыдущих эксперементов лучшими параметрами для реализуемой модели являются такие гиперпараметры:k=4, косинусная метрика, алгоритм *«brute»* и использование весов. В ходе этого эксперемента была измерена точность классификации для тестовой выборки, на основе обучающей. Точность составляет 0.9752, что говорит о том, что модель не переобучена, так как на кросс-валидицаии точность составляет 0.9741.

Для сравнения моего результата можно обратиться на Kaggle. Первые 50 лидерующих алгоритмов добились точности 1.0. Для выяснения ошибок моей модели была построена матрица ошибок(Рис.4).

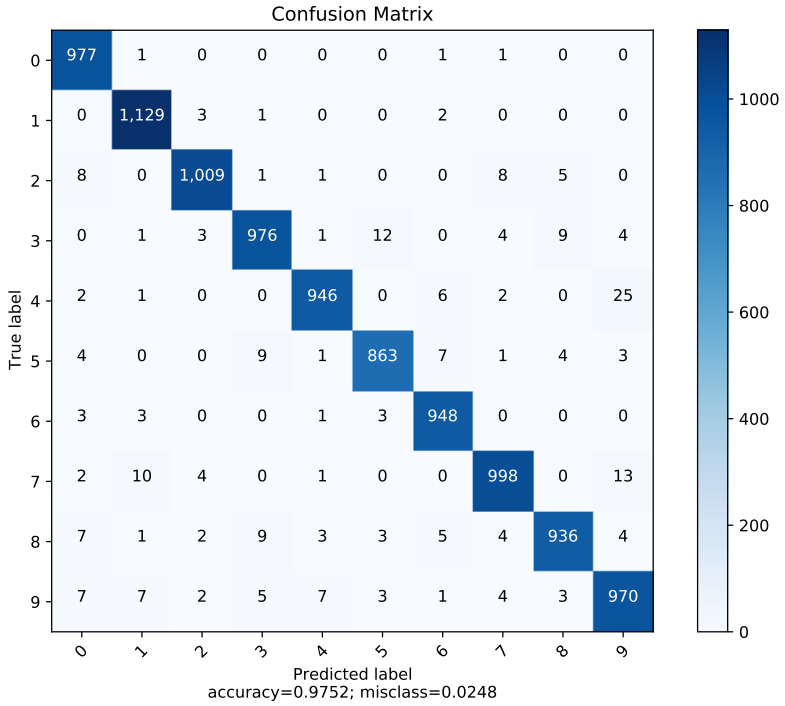
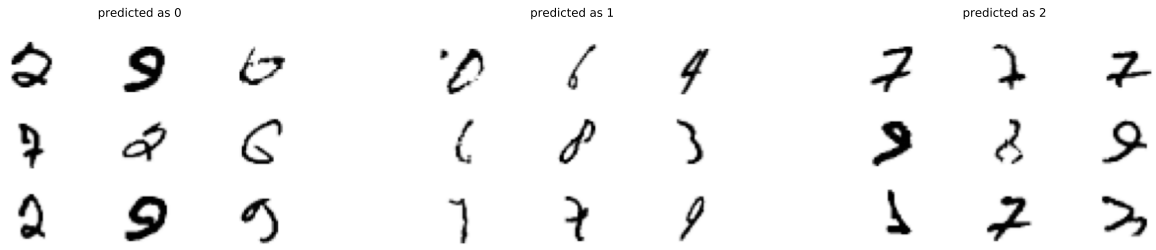


Рис.4 Матрица ошибок для классификации тестовых объектов

Согласно графику, неправильно классифицированные данные имеют закономерность. Чаще всего ошибка допускалась в определении числа ’7’. Модель предсказывала ’9’ или ’1’ вместо ’7’. Это связанно с наклоной чертой и окружностью сверху у девятки. Также возникли проблемы с рапознованием ’4’, 25 раз модель предсказала ’9’ вместо ’4’, это самая частая ошибка в данном эксперементе. Ниже преведены примеры в которых модель предсказала неправильный результат.



3.5 Размножение обучающей выборки

Цель данного эксперемента заключается в размножении обучающей выборки с помощью поворотов, смещения и применений гауссовского фильтра.

1. Величина поворота: 5, 10, 15(в каждую из двух сторон)
2. Величина смещения: 1, 2, 3 пикселя (по каждой из двух размерностей)
3. Дисперсия фильтра гауса: 0.5, 1, 1.5

3.5.1 Поворот изображения

Результаты описанного эксперемента для поворотов изображений (повороты в каждую из сторон.(Табл. 4)

Табл. 1 Точность предсказания в зависимости от поворота угла

углы поворотов	точность предсказания
-5	0.9744
5	0.9752
-10	0.9755
10	0.9762
-15	0.9741
15	0.9758

Максимальная точность на кросс-валидации была достигнута при повороте на 10 градусов.Применив этот механизм к обучающей выборке получаем точность 0.9789. На матрице ошибок(Рис. 5) видно, что благодаря повороту изображений уменьшилось количество ошибок в распознавании цифр, зависящих от степени наклона и содержащие наклоные элементы: 7, 1, 5, 9.

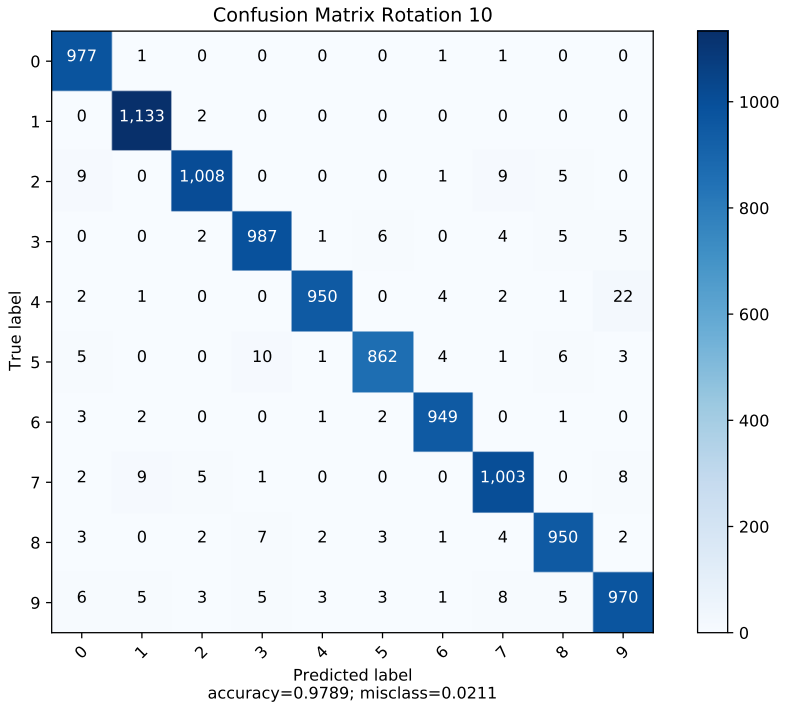


Рис.5 Матрица ошибок после добавления объектов с изменением угла поворота

3.5.2 Смещения

Результаты описанного эксперемента для смещения изображений.(Табл. 2)

Табл. 2 Точность предсказания в зависимости от смещения изображений

смещение(в каждую из 2 сторон)	точность предсказания
на 1 пиксель	0.9765
на 2 пикселя	0.9751
на 3 пикселя	0.9746

Максимальная точность на кросс-валидации была достигнута при смещении на 1 градус вправо. Применив этот механизм к обучающей выборке получаем точность 0.9759. На матрице ошибок(Рис. 6) видно, что благодаря этому смещению точность увеличилась, однако закономерностей не наблюдается. То есть, увеличилось распознование во всех классах объектов.

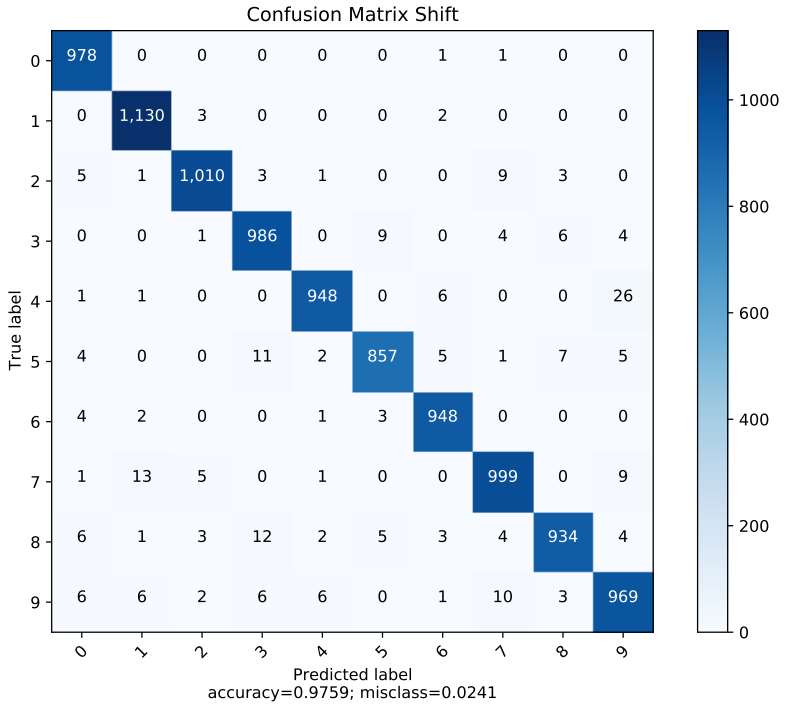


Рис.6 Матрица ошибок после добавления смещенных объектов

3.5.3 Дисперсия фильтра Гаусса

Результаты описанного эксперемента для применения фильтра Гаусса.(Табл. 3)

Табл. 3 Точность предсказания в зависимости от фильтра Гаусса

значение параметра	точность предсказания
0.5	0.9754
1	0.9761
1.5	0.9763

Максимальная точность на кросс-валидации была достигнута при значении параметра 1.5. Применив этот механизм к обучающей выборке получаем точность 0.9762. На матрице ошибок(Рис. 7) видно, что благодаря гауссовскому фильтру уменьшилось число ошибок у чисел, написанных нечетко, имеющих утолщенные линии.

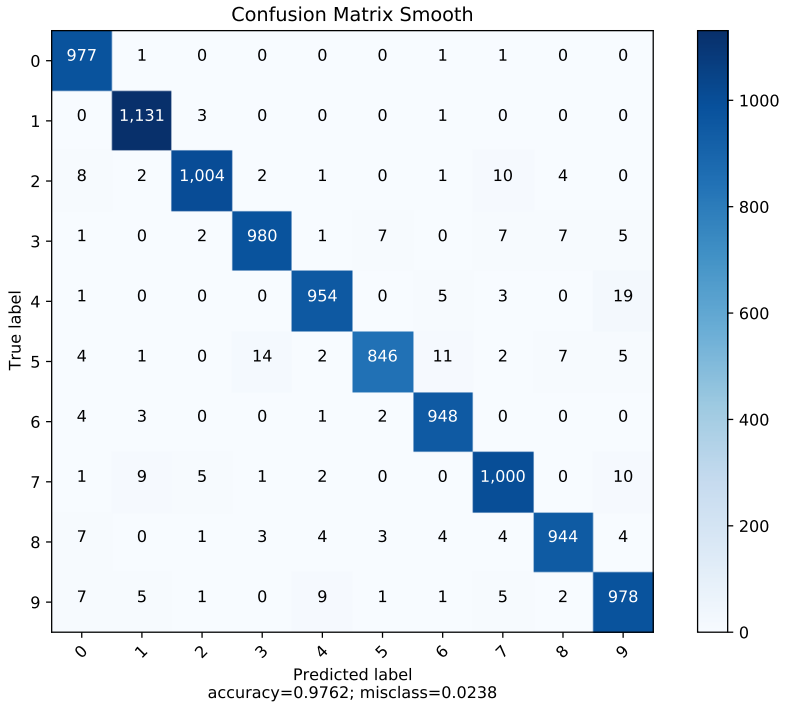


Рис.7 Матрица ошибок после добавления объектов с Фильтром Гаусса

3.6 Размножение тестовой выборки

Реализуем тот же эксперемент, что и прошлый, однако будем преобразовывать тестовую выборку. Используем кросс-валидацию с 3-мя фолдами для подбора параметров. Каждому объекту валидационной выборки сопоставим множество, состоящее из объектов, полученных путем применения рассматриваемых преобразований к исходному. Расстояние от множества до объекта обучающей выборки будем считать минимум расстояния до этого объекта по всем элементам множества.

3.6.1 Поворот изображения

Результаты описанного эксперемента для поворотов изображений (повороты в каждую из сторон.(Табл. 4)

Табл. 4 Точность предсказания в зависимости от поворота угла

углы поворотов	точность предсказания
-5	0.9764
5	0.9772
-10	0.9725
10	0.9715
-15	0.9751
15	0.9738

Максимальная точность на кросс-валидации была достигнута при повороте на 10 градусов.Применив этот механизм к тестовой выборке получаем точность 0.9783.

Сравнив данные с проведением данного опыта на обучающей выборке, делается вывод, что точность упала, при этом есть и те данные, на которые делают ошибки оба алгоритма. Таким образом, алгоритм размножения поворотом лучше применять для обущающих данных.

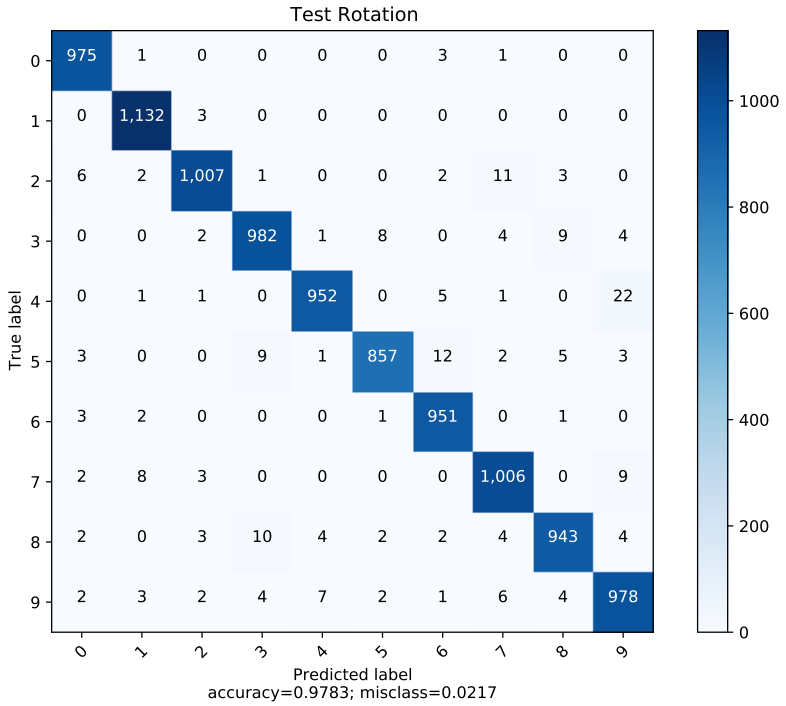


Рис.8 Матрица ошибок после поворота тестовых объектов

3.6.2 Фильтр Гаусса

Результаты описанного эксперемента для применения фильтра Гаусса.(Табл. 5)

Табл. 5 Точность предсказания в зависимости от фильтра Гаусса

значение параметра	точность предсказания
0.5	0.9726
1	0.971
1.5	0.9684

Максимальная точность на кросс-валидации была достигнута при значении параметра 0.5. Применив этот механизм к тестовой выборке получаем точность 0.9671. На матрице ошибок(Рис. 9) видно, что точность прогноза ухудшилась. Однако, есть данные, которые были предсказаны правильно при увелечении тестовой выборки, а не обучающей. Это стали изображения с лишними штрихами, пропадающими при размытии. Однако применение фильтра к обучающей выборке помогает правильно распознать цифры, изображенные слишком жирными линиями.

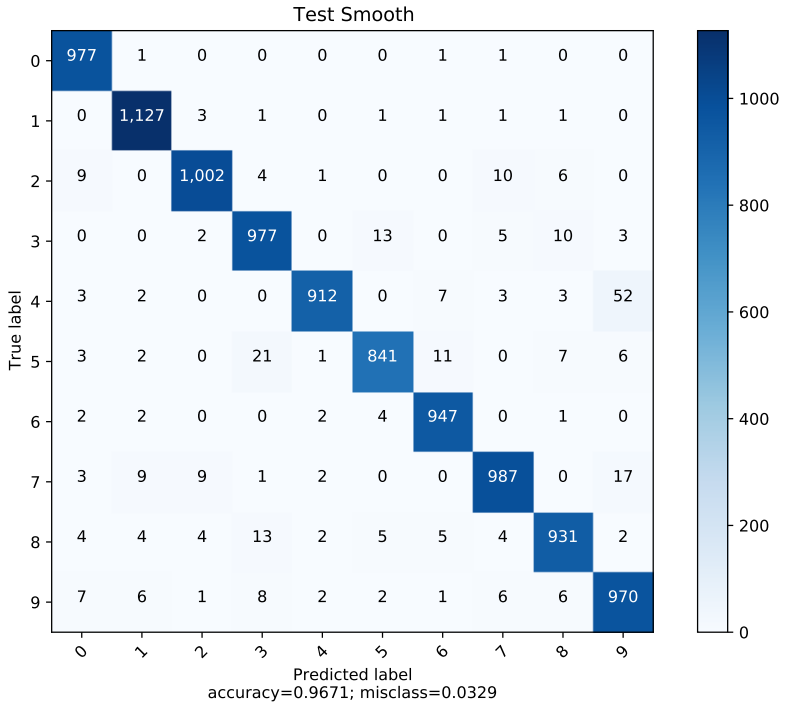


Рис.9 Матрица ошибок после добавления в тестовую выборку объектов с Фильтром Гаусса

3.6.3 Смещения

Результаты описанного эксперемента для смещения изображений.(Табл. 6)

Табл. 6 Точность предсказания в зависимости от смещения изображений

смещение(в каждую из 2 сторон)	точность предсказания
на 1 пиксель	0.9802
на 2 пикселя	0.9772
на 3 пикселя	0.9746

Максимальная точность на кросс-валидации была достигнута при смещении на 1 градус вправо. Применив этот механизм к обучающей выборке получаем точность 0.9803. На матрице ошибок(Рис. 8) видно, что смещение обучающей выборки дает похожие показатели, как и смещение тестовых данных.

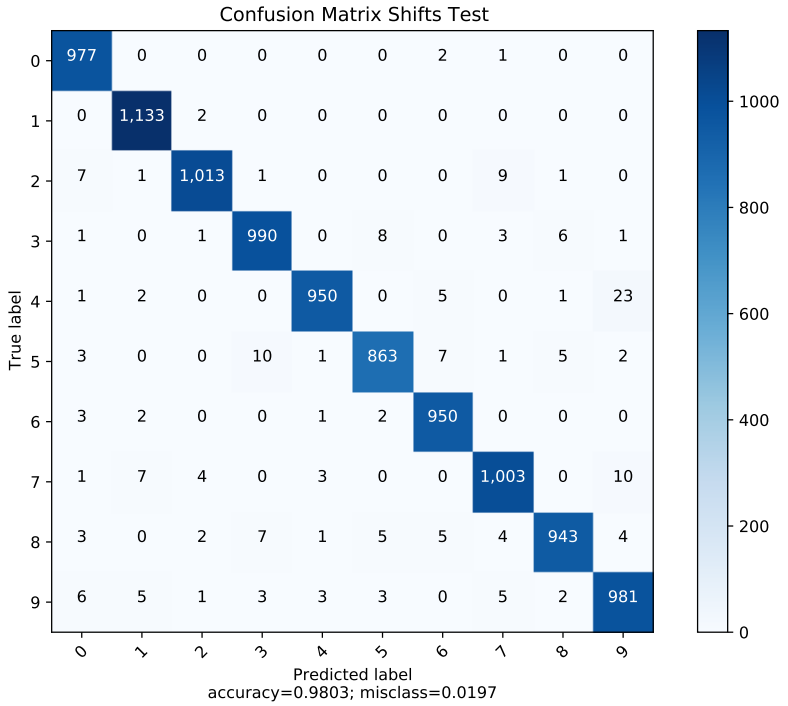


Рис.8 Матрица ошибок после добавления смещенных объектов

Применение данного механизма привело к наилучшему показателю точности, среди всех эксперементов. Это говорит о том, что в модели нет больших проблем с рапознованием какой-то конкретной подвыборки, и общее незначительное улучшение данных дает наивысшую точность распознованиня, чем точечные улучшения методами поворота и фильтра Гаусса.

3.7 Вывод

Проведенные эксперемнты показали, что метрический алгоритм классфикации методом k-ближайших соседей дает хорошую точность на датасете изображений цифр MNIST. Однако основной проблемой метода является низкая скорость работы($O(ND)$), где N - число объектов обучающей выборки, D - число признаков.

Эксперементы по размножению тестовых и обучающий данных показывают, что нужно смотреть на интерпретируемые данные. Зачастую лучше обойтись размножением тестовых данных, так как это намного быстрее. Также нужно учитывать проблемные данные, возможно применение узконапрвенного алгоритма обработки данных может привести к повышению точности распознования.