

– при возрастании интенсивности СПП при гипотезе на 2 % для режима ПК – увеличение ошибок 1-го рода в 32,7 и 195,2 раз; для режима ОК – уменьшение ошибок 1-го рода в $7,4 \cdot 10^{-3}$ и $6,1 \cdot 10^{-4}$ раз.

Очень важный практический вывод при решении задачи обнаружения слабого оптического сигнала, который в открытой литературе не встречается. Если в реальном ПОК-процессоре, построенном на базе ПОК-алгоритма, наблюдение за объектом идет при ПК и интенсивность фона, на котором он наблюдается, возросла, то, если есть возможность, надо переходить в режим ОК. В этом случае искусственное увеличение интенсивности фона только облегчает процесс обнаружения.

Подчеркнем, ПОК-процессор в описанном варианте использования имеет свойства не просто устойчивости (как известный ПУЛТ-процессор), а улучшения показателей качества за счет внешнего воздействия.

В доступной авторам литературе пока такому свойству не дано определение?

В связи с этим для неопределенных условий положительного и отрицательного контрастов ПОК-процессор оказывается предпочтительнее применения более сложного ПУЛТ-процессора, обеспечивающего устойчивость за счет адаптации порога обнаружения.

УДК 004.89

Н. В. Пацей, доц., канд. техн. Наук;

И.Г. Сухорукова, ст. преп.; О. Л. Панченко, ассист. (БГТУ, г. Минск)

МОДЕЛЬ КЛАССИФИКАЦИИ ОБЪЕКТОВ ИЗОБРАЖЕНИЙ ПО НЕСКОЛЬКИМ МЕТКАМ И КЛАССАМ

Некоторые задачи классификации требуют прогнозирования более чем одной метки класса. Это означает, что метки класса или принадлежность классу не исключают друг друга. Задача классификации помечает каждый образец x метками из возможных классов $n_classes$, где x может быть от 0 до $n_classes$ включительно. Формально двоичный вывод назначается каждому классу для каждой выборки. Многозначная или классификация по нескольким меткам обрабатывает несколько классов одновременно, учитывая коррелированное поведение между ними.

Например, изображение может содержать один тематический класс объектов (корабль, лес и т. д), а могут несколько одновременно (дом и поле). Количество меток (свойств) и количество классов для каждой метки может быть больше 2-х (рис. 1).

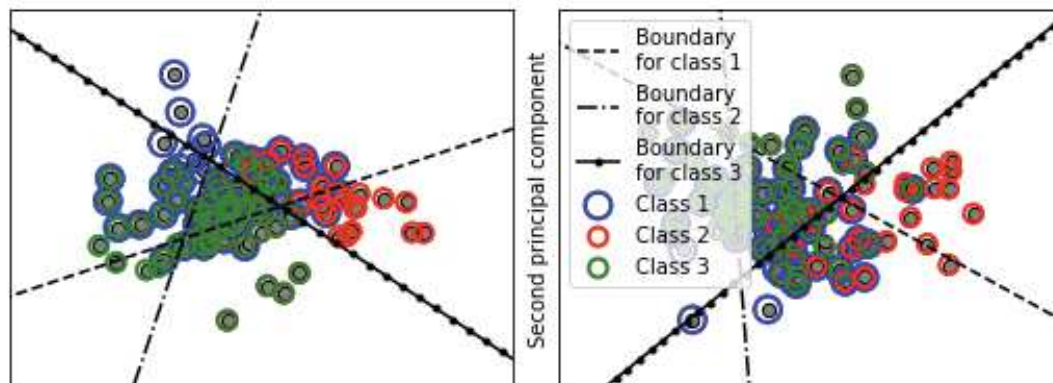


Рисунок 1 – Графическая интерпретация классификации по нескольким меткам для трех классов

Процесс классификации в этом случае предполагает подготовку данных, построение модели, прогнозирование и проверку точности. Программная модель разрабатывалась на языке *Python* (*Jupyter notebook*) на основе библиотек: *Scikit-learn*, *SciPy*, *Pandas*, *Matplotlib*, *NumPy*. Модель настраивалась с помощью класса *Sklearn.MultiOutputClassifier*. В качестве оценщика был реализован метод на основе *ECOC* (Error Correcting Output Codes) [1].

При исследовании использовались синтетические наборы данных с характеристиками, значения которых представлены в таблице. Количество признаков для всех наборов равно 10.

Таблица – Характеристики исследуемых наборов данных

| Номер набора | Количество классов, n classes | Количество меток, n labels | Количество образцов в наборе, n samples |
|--------------|---------------------------------|------------------------------|---|
| 1 | 3 | 2 | 1000 |
| 2 | 5 | 3 | 1000 |
| 3 | 7 | 2 | 500 |
| 4 | 10 | 5 | 500 |

Фрагмент входных и выходных значений набора 3 с 7-ю классами и 10-ю признаками представлен на рис. 2. Единица в соответствующем столбце означает присутствие метки.

Для метаоценки решения задач многозначной классификации использовались: средняя точность, матрица неточностей (confusion matrix), площадь под кривой рабочих характеристик приемника (Area Under the Receiver Operating Characteristic Curve – *ROC AUC*) по оценкам прогнозов, точность (*Precision*) и полнота (*Recall*) для классов, F-мера (*F-score*).

```

[8. 6. 3. 5. 3. 8. 0. 6. 6. 8.] => [0 1 0 1 0 1 1]
[3. 2. 4. 8. 4. 3. 2. 3. 3. 1.] => [0 0 0 0 0 0 0]
[ 5. 5. 5. 3. 3. 6. 0. 2. 3. 11.] => [1 1 0 1 0 0 0]
[1. 0. 2. 7. 3. 5. 5. 2. 5. 5.] => [1 1 0 1 1 1 0]
[1. 2. 2. 9. 8. 7. 7. 4. 9. 1.] => [0 0 0 0 0 1 0]
[ 8. 3. 7. 4. 11. 5. 0. 7. 6. 3.] => [1 1 0 1 1 1 1]
[ 4. 4. 16. 7. 4. 5. 2. 2. 5. 12.] => [1 1 0 1 1 0 0]
[4. 5. 3. 5. 9. 5. 0. 6. 1. 6.] => [1 1 0 0 0 0 0]
[ 5. 7. 10. 5. 4. 7. 3. 6. 5. 5.] => [0 0 0 0 0 0 0]
[ 5. 0. 3. 4. 1. 4. 1. 3. 4. 12.] => [1 0 0 0 0 0 0]

```

Рисунок 2 – Фрагмент входных и выходных значений набора данных

Отчет по метрикам для *ECOC_KNeighborsClassifier* с *code_size* = 1.5 для 3-х классов и 2-х меток выглядит как показано на рис. 3.

В ходе проведенных исследований установлено, что ни изменение скорости кода и других параметров *ECOC*, ни изменение алгоритмов оценки классификации не привели к изменению значений точности и полноты.

```

score: 0.7533333333333333
ROC AUC: 0.8392752631763134
ROC AUC y0 : 0.8689
ROC AUC y1 : 0.8109
ROC AUC y2 : 0.8380
Confusion matrix y: 0
[[53 10]
 [ 9 78]]
Confusion matrix y: 1
[[ 27 14]
 [ 4 105]]
Confusion matrix y: 2
[[111  6]
 [ 9 24]]
Classification_report y: 0
      precision    recall  f1-score   support

     0       0.85     0.84     0.85         63
     1       0.89     0.90     0.89         87

 accuracy          0.87         150
 macro avg         0.87     0.87     0.87         150
 weighted avg     0.87     0.87     0.87         150

Classification_report y: 1
      precision    recall  f1-score   support

     0       0.87     0.66     0.75         41
     1       0.88     0.96     0.92        109

 accuracy          0.88         150
 macro avg         0.88     0.81     0.84         150
 weighted avg     0.88     0.88     0.87         150

Classification_report y: 2
      precision    recall  f1-score   support

     0       0.93     0.95     0.94         117
     1       0.80     0.73     0.76          33

 accuracy          0.90         150
 macro avg         0.86     0.84     0.85         150
 weighted avg     0.90     0.90     0.90         150

```

Рисунок 3 – Формат метрик оценки для классификатора *ECOC_KNeighborsClassifier*

Таким образом, модель дает достаточно высокую точность классификации, но применение выходных кодов коррекции ошибок не дает существенных преимуществ для многоклассовой классификации по нескольким меткам.

ЛИТЕРАТУРА

1. Пацей Н.В, Самаль А.Д., Годун А.В. Алгоритм многоуровневой классификации объектов изображений на основе Error Correcting Output Codes // Информационные технологии: материалы 84-й науч.-техн. конференции профессорско-преподавательского состава. научных сотрудников и аспирантов (с международным участием) Минск, 2020 – С. 76–78.

УДК 004.724.4

Д.В. Шиман, доц., канд. техн. наук; Д.В. Котович, магистр
(БГТУ, г. Минск)

АЛГОРИТМ БАЛАНСИРОВКИ С ИЗМЕНЯЮЩИМИСЯ ВЕСОВЫМИ КОЭФФИЦИЕНТАМИ СЕРВЕРОВ И ПОРОГОВЫМ ЗНАЧЕНИЕМ

Разработанный алгоритм балансировки базируется на известном алгоритме Weighted Round Robin [1, 2]. Изначально при запуске балансировщика указываются весовые коэффициенты для каждого из серверов.

На каждом из серверов сохраняется время обработки последней тысячи запросов, в последствии каждые 2 минуты балансировщик делает запрос по определенному адресу для получения среднего времени обработки запроса от сервера в миллисекундах. Визуализация этого процесса изображена на рисунке 1.

Также дополнительно благодаря этому запросу балансировщик будет узнавать, находится ли сервер в рабочем состоянии, тем самым в случае, если он не работоспособен, балансировщик при выборе следующего сервера не будет его даже учитывать. После получения значений времени от серверов, для каждого из серверов среднее время преобразуется в коэффициент балансировки рассчитывается по формуле:

$$k = \frac{t}{100}, \quad (1)$$

где t – среднее время обработки последней тысячи запросов.