Ушаков Максим,

ФКН, НоД, ИССА, 1 курс

**Описание данных:**

**1. Оценка доступности машин (car.data):**

имеем 6 порядковых признаков:

* стоимость автомобиля {low - vhigh}
* стоимость обслуживания {low - vhigh}
* количество дверей {2 - 5}
* вместимость (по пассажирам) {2, 4, more}
* размер багажника {small - big}
* надёжность автомобиля {low - high}

Целевой признак:

Доступность автомобиля: {unacc, acc, good, v-good}

Признаки были переведены в качественные следующим образом:

где - значение признака меньше либо равно ,

- значение признака больше либо равно ,

остальные – аналогично.

Преобразование данных проводилось самой программой (*main.py, read\_data2()*).

**2. Крестики - нолики (connect-4.data):**

Имеем признака, принимающих значения .

Целевой признак – результат игры: .

Данные преобразовывались следующим образом:

1) Сначала значимая часть переносилась в левый нижний угол:

Таким образом, верхние строки и правые столбцы оставались незаполненными, поэтому они не рассматривались (в итоге матрицу 6 на 7 удалось привести к матрице 4 на 5). Тем самым исключались лишние атрибуты, вследствие чего алгоритм работал быстрее (общее число признаков значительно уменьшалось).

Признак «элемент j-ой ячейки» разбивался на два признака:

На этом заканчивалась предварительная обработка данных (*main.py, read\_data1()*).

**Описание алгоритма:**

1. Исходные данные произвольным образом делятся на обучающую и тестовую выборки, в соотношении 7:3.

2. Полученная обучающая выборка делится на классы в соответствии со значением целевого признака (для car.data получаем 4 класса, для connect-4.data получаем 3 класса).

3. Для каждого класса строим решётку формальных понятий.

**Далее имеем два алгоритма построение гипотез и их агрегирования:**

**1 алгоритм: (classify\_test\_set2(self, max\_part\_contradictions=0, min\_part\_support=0.001, threshold=0))**

4. Передаём в функцию классификации минимальную поддержку и максимальный процент контрпримеров .

5. Для каждого формального понятия :

Если и количество контрпримеров не превосходит , тогда - гипотеза.

6. Для каждого тестового примера определяем, каким гипотезам он удовлетворяет:

7. Для каждого класса считаем суммарный вес всех гипотез, как объединение их объёмов:

8. Сравнивается относительный вес определённого класса и доля объектов данного класса среди общего числа объектов:

9. Среди всех выбирается максимальный, после чего он сравнивается с заданным пороговым значением . Если больше порогового значения, то объекту присваивается метка , в противном случае ему присваивается метка .

Примечание: максимальное значение будет заведомо больше 0, поэтому и порог выбирается больше 0.

**2 алгоритм: (classify\_test\_set1(self, max\_num\_contradictions=0, min\_support=1, threshold=0.51))**

4. Передаём в функцию классификации максимальное количество контрпримеров .

5. Для каждого формального понятия :

Если количество контрпримеров не превосходит , тогда - гипотеза.

6. Для каждого тестового примера определяем, каким гипотезам он удовлетворяет:

7. Для каждого класса считаем количество гипотез, которым удовлетворяет тестовый объект :

8. Среди всех выбирается максимальный, после чего его относительное значение сравнивается с заданным пороговым значением . Если больше порогового значения, то объекту присваивается метка , в противном случае ему присваивается метка .

**Описание мер качества:**

Поскольку в обоих наборах данных целевых признаков (классов) не два, а несколько, то рассматривать ошибки первого и второго рода смысла нет (во-первых, слишком много ошибок придётся рассматривать, во-вторых, построить ROC-кривую всё равно не удастся).

**Рассмотрим две меры качества:**

1. где - количество правильно классифицированных объектов класса , – общее количество классифицированных объектов.

2. , где - доля классифицированных объектов, – множество объектов тестовой выборки.

**Динамика изменения мер качества в зависимости от выбранных параметров:**

**1 алгоритм:**

Имеем три параметра:

– максимальная доля контрпримеров,

– минимальная поддержка,

- пороговое значение

Таблица изменения мер качества при увеличении следующих параметров (один параметр меняется, остальные остаются зафиксированными):

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Если установить на нуле, и менять значения двух других параметров, сохраняя уровень на 94%, то можно получить следующие значения точности:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| 0 | 0,001 | 0 | 97% | 94% |
| 0,025 | 0,01 | 0 | 97% | 94% |
| 0,05 | 0,06 | 0 | 95% | 94% |
| 0,075 | 0,1 | 0 | 92% | 94% |
| 0,1 | 0,14 | 0 | 90% | 94% |
| 0,125 | 0,17 | 0 | 85% | 94% |
| 0,15 | 0,2 | 0 | 82% | 94% |

Как видно из таблицы, локальных максимумов для точности нет, и она убывает с ростом поддержки и максимальной допустимой доли контрпримеров. Поэтому оптимальным значением для поддержки считаем . Проводя аналогичный анализ для и , получаем, что оптимальные значения для этих параметров равны и (разница в десятых долях процента, поэтому с учётом округления лучшие значения метрик мы всё равно получаем равными и ).

**2 алгоритм:**

Имеем два параметра:

– максимальное число контрпримеров,

- пороговое значение

Таблица изменения мер качества при увеличении следующих параметров (один параметр меняется, остальные остаются зафиксированными):

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Сохраняя уровень на 92% и меняя значения двух параметров, можно получить следующие значения точности:

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 0 | 0.51 | 92% | 92% |
| 2 | 0.54 | 95% | 92% |
| 4 | 0.57 | 90% | 92% |
| 6 | 0.62 | 87% | 92% |
| 8 | 0.69 | 65% | 92% |
| 10 | 0.77 | 37% | 92% |

Как видно из таблицы, оптимальные значения параметров:

Сравнивая два алгоритма между собой, первый алгоритм даёт лучшие результаты точности и доли классифицируемых объектов.

**Примечание:**

Здесь приведены сравнения мер качества для разных входных параметров для данных *'car.data'*. Для *'connect-4.data'* были проведены аналогичные исследования, однако в этом случае дисперсия мер качества (при одних и тех же входных данных) слишком высока, чтобы можно было проводить хорошее сравнение.