Работа № 3.4 Наивный байесовский классификатор

1. Теоретические сведения

Метод классификации, основанный на наивном байесовском классификаторе, является алгоритмом обучения с учителем, в котором применяется теорема Байеса со строгим (наивным) предположением о независимости между каждыми парами признаков. Предположение о независимости позволяет избавиться от сложной схемы оценки параметров классификатора. Это позволяет применять алгоритм на больших выборках. Также классификация оказывается достаточно точной: недостаточной для высокоточных систем классификации, однако удовлетворительной для грубой оценки и сравнения с другими алгоритмами.

Вероятностная модель для классификатора – это условная модель $P(y \mid x_1, \dots, x_n)$ над независимой переменной класса y и признаками $x_1, \dots, x_n$ по теореме Байеса

$$P(y \mid x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n \mid y)}{P(x_1, \dots, x_n)}$$

и предположении независимости, получаем:

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y),$$

Для всехi, это соотношение упрощается

$$P(y \mid x_1, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \dots, x_n)}$$

Поскольку $P(x_1, \dots, x_n)$ является постоянной величиной с учетом входных данных, мы можем использовать следующее правило классификации:

$$P(y \mid x_1, \dots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i \mid y)$$

$$\Downarrow$$

$$\hat{y} = arg\max_{y} P(y)\prod_{i=1}^{n} P(x_i \mid y),$$

и использовать оценку апостериорного максимума для оценки $P(y)$ и $P(x_i \mid y)$. Различные наивные байесовские классификаторы различаются, в основном, допущениями, которые они делают относительно $P(x_i \mid y)$.

Причина, по которой наивные байесовские модели столь эффективны, заключается в том, что они оценивают параметры, рассматривая каждый признак отдельно и по каждому признаку собирают простые статистики классов.

Положительные и отрицательные стороны наивного байесовского алгоритма

Положительные стороны:

Классификация, в том числе многоклассовая, выполняется легко и быстро.

Когда допущение о независимости выполняется, НБА превосходит другие алгоритмы, такие как логистическая регрессия (logistic regression), и при этом требует меньший объем обучающих данных.

НБА лучше работает с категорийными признаками, чем с непрерывными. Для непрерывных признаков предполагается нормальное распределение, что является достаточно сильным допущением.

Отрицательные стороны:

Если в тестовом наборе данных присутствует некоторое значение категорийного признака, которое не встречалось в обучающем наборе данных, тогда модель присвоит нулевую вероятность этому значению и не сможет сделать прогноз. Это явление известно под названием «нулевая частота» (zerofrequency). Данную проблему можно решить с помощью сглаживания. Одним из самых простых методов является сглаживание по Лапласу (Laplacesmoothing).

Хотя НБА является хорошим классификатором, значения спрогнозированных вероятностей не всегда являются достаточно точными. Поэтому не следует слишком полагаться на результаты, возвращенные методом *predict_proba*.

Еще одним ограничением НБА является допущение о независимости признаков. В реальности наборы полностью независимых признаков встречаются крайне редко.

Приложения наивного байесовского алгоритма

Классификация в режиме реального времени. НБА очень быстро обучается, поэтому его можно использовать для обработки данных в режиме реального времени.

Многоклассовая классификация. НБА обеспечивает возможность многоклассовой классификации. Это позволяет прогнозировать вероятности для множества значений целевой переменной.

Классификация текстов, фильтрация спама, анализ тональности текста. При решении задач, связанных с классификацией текстов, НБА превосходит многие другие алгоритмы. Благодаря этому, данный алгоритм находит широкое применение в области фильтрации спама (идентификация спама в электронных письмах) и анализа тональности текста (анализ социальных медиа, идентификация позитивных и негативных мнений клиентов).

Рекомендательные системы. Наивный байесовский классификатор в сочетании с коллаборативной фильтрацией[2] (collaborative filtering) позволяет реализовать рекомендательную систему. В рамках такой системы с помощью методов машинного обучения и интеллектуального анализа данных новая для пользователя информация отфильтровывается на основании спрогнозированного мнения этого пользователя о ней.

2. Задача: Рубрикация новостных статей

Имеется коллекция новостных статей

$$D = \{d_1, d_2, \ldots, d_n\}$$

Имеется множество рубрик новостей

$$Y = \{y_1, y_2, \ldots, y_k\}$$

Необходимо построить модель, которая для заданной статьи определяет, к какой рубрики она относится

Векторное представление текстов

Словарь W – множество слов (после предобработки, нормализации, удаления стоп-слов), $|W| = m$

Документы D – множество статей, $|D| = n$

Статья x представляется как вектор

$$x = (x_1, x_2 \ldots, x_m)$$

Document-termmatrix$X$: строки – документы, столбцы – слова

[2]**Коллаборативная фильтрация** (англ. *collaborative filtering*)—это один из методов построения прогнозов (рекомендаций) в рекомендательных системах, использующий известные предпочтения (оценки) группы пользователей для прогнозирования неизвестных предпочтений другого пользователя.

Элементы матрицыx_{dt}

- Бинарные: $\begin{cases}1, t \in d\\0, t \notin d\end{cases}$
- Termfrequency (tf): сколько раз слово *t* встречается в документе*d* или производная от этого величина
- TF-IDF: $tfidf(t,d,D) = tf(t,d) \cdot idf(t,D)$,

$$idf(t,D) = log\frac{n}{1+|\{d \in D: t \in d\}|}$$

3. Байесовский классификатор

Задача: необходимо найти наиболее вероятное значение $y \in Y$ класса объекта $x = (x_1, x_2 \ldots, x_m)$ при условии заданных признаков объекта.

$$y(x) = \underset{y \in Y}{argmax} p(y|x)$$

Формула Байеса

апостериорная вероятность y при условии x $p(x|y)$ — *априорная вероятность класса y* $p(y)$

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}$$

априорная вероятность объекта x $p(x)$

Максимум вероятности:

$$y(x) = \underset{y \in Y}{argmax} p(x|y)p(y)$$

Часто по умолчанию предполагают значения классов равновероятными

$$y(x) = \underset{y \in Y}{argmax} p(x|y)$$

Рубрикация новостных статей

$$y(x) = \underset{v \in Y}{argmax} p(x_1 = a_1, x_2 = a_2, \ldots, x_m = a_m|y = v)p(y = v)$$

Оценка $p(y = v)$: частота встречаемости статей рубрики *v* в коллекции $p(x_1 = a_1, x_2 = a_2, \ldots, x_m = a_m|y = v)$ - вероятность в точности такого набора слов. Оценить невозможно, т.к. нет такой статистики

Наивный Байес

Предположение об условной независимости атрибутов в общем случае неверно. Но эти зависимости совпадают для разных классов и «сокращаются» при оценке вероятностей. Например, грамматические зависимости остаются неизменными для всех рубрик новостей.

Наивный Байес хорошо показывает себя при решении задачи классификации, однако он не всегда может быть пригоден для оценки вероятностей.

Предположим условную независимость атрибутов при условии данного значения целевой функции

$$p(x_1 = a_1, x_2 = a_2, \ldots, x_m = a_m|y = v) =$$
$$= p(x_1 = a_1|y = v)p(x_2 = a_2|y = v) \ldots p(x_m = a_m|y = v)$$

Генеративная модель

Модель, по которой порождается документ:

MultinomialNaïveBayes: | Документ – это последовательность событий. Каждое событие –

BernoulliNaïveBayes:	это случайный выбор одного слова из словаря Документ – это вектор бинарных атрибутов, показывающих, встретилось ли в документе то или иное слово

Классификатор с мультиномиальным распределением (Multinomial Naïve Bayes)

Используется в случае дискретных признаков. Например, в задаче классификации текстов признаки могут показывать, сколько раз каждое слово встречается в данном тексте.

Пусть x_i - частота встречаемости слова *i* в документе

Тогда вероятность слова *i* в документе класса *v* оценивается как

$$\hat{\theta}_{vi} = \frac{N_{vi} + \alpha}{N_v + \alpha m}, где$$

$N_{vi} = \sum_{x:x\in D, y(x)=v} x_i \qquad N_v = \sum_{i=1}^{m} N_{vi} \alpha \geq 0$

Классификация:

$$y(x) = \underset{v\in Y}{argmax} \prod_{i=1}^{m} \hat{\theta}_{vi}^{xi}\, p(y = v)$$

Классификатор с распределением Бернулли (Bernoulli Naïve Bayes)

Используется в случае двоичных дискретных признаков (могут принимать только два значения: 0 и 1). Например, в задаче классификации текстов с применением подхода «мешок слов» (bagofwords) бинарный признак определяет присутствие (1) или отсутствие (0) данного слова в тексте.

Пусть $x_i \in \{0,1\}$ - встречается (1) или нет (0) слово *i* в документе

Тогда вероятность слова *i* в документе класса *v* оценивается как

$$P(x_i|y = v) = \frac{1 + N_{vi}}{2 + |\{x: x \in D, y(x) = v\}|}, где$$

$$N_{vi} = \sum_{x:x\in D, y(x)=v} x_i$$

Классификация:

$$y(x) = \underset{v\in Y}{argmax} p(y = v) \prod_{i=1}^{m} (p(x_i|y = v)a_i + (1 - p(x_i|y = v))(1 - a_i))$$

Особенности применения BernoulliNB и MultinomialNB

BernoulliNB принимает бинарные данные, MultinomialNB принимает счетные или дискретные данные (то есть каждый признак представляет собой подсчет целочисленных значений какой-то характеристики, например, речь может идти о частоте встречаемости слова в предложении). BernoulliNB и MultinomialNB в основном используются для классификации текстовых данных.

MultinomialNB и BernoulliNB имеют один параметр alpha, который контролирует сложность модели. Параметр alpha работает следующим образом: алгоритм добавляет к данным зависящее от alpha определенное количество искусственных наблюдений с положительными значениями для всех признаков. Это приводит к «сглаживанию» статистик. Большее значение alpha означает более высокую степень сглаживания, что приводит к построению менее сложных моделей. Алгоритм относительно устойчив к разным значениям alpha. Это означает, что значение alpha не оказывает значительного влияния на хорошую работу модели. Вместе с тем тонкая настройка этого параметра обычно немного увеличивает правильность.

4. Пример: 20 Newsgroups[3]

URL: http://qwone.com/~jason/20Newsgroups/

- Набор новостных статей «20 Newsgroups»
- 18000 новостных статей из 20 различных рубрик.

РешениевScikit-learn

```
from sklearn.datasetsimport fetch_20newsgroups
from pprintimport pprint
from sklearn.feature_extraction.textimport CountVectorizer
from sklearn.naive_bayesimport MultinomialNB
from sklearn.naive_bayesimport BernoulliNB
from sklearnimport metrics
from sklearn.metricsimport classification_report

# Загрузка данных для обучения и тестирования
newsgroups_train = fetch_20newsgroups(subset='train',
remove=('headers', 'footers', 'quotes'))
newsgroups_test = fetch_20newsgroups(subset='test')
news = fetch_20newsgroups(subset='all')

# Список новостных рубрик
# (совпадает в обучающей  и тестовой выборке)
print("Число наблюдений в обучающей выборке\n",
newsgroups_train.filenames.shape)
print("Число наблюдений в тестовой выборке\n",
newsgroups_test.filenames.shape)
print("Список новостных рубрик\n")
pprint(list(newsgroups_train.target_names))
pprint(list(newsgroups_test.target_names))
# Приведение данных к document-term матрице
vectorizer = CountVectorizer()
sparse_train = vectorizer.fit_transform(newsgroups_train.data)
sparse_test = vectorizer.transform(newsgroups_test.data)
# Размерность данных (в dense и sparse совпадает)
print("Размерность обучающей выборки sparse\n", sparse_train.shape)
print("Размерность тестовой выборки sparse\n", sparse_test.shape)
dense_train = sparse_train#.toarray()
dense_test = sparse_test#.toarray()
print("Размерностьобучающейвыборки dense\n", dense_train.shape)
print("Размерность тестовой выборки dense\n", dense_test.shape)
mnb = MultinomialNB(alpha= 1)
for iin range(0, 100):
# Используйте данные обучения для оценки параметров модели и
прогнозирования
mnb.fit(sparse_train, newsgroups_train.target)
pred = mnb.predict(sparse_test)
# Точностьклассификации
a = metrics.accuracy_score(newsgroups_test.target, pred, normalize=True)
print("Точность классификации – доля верно классифицированных объектов из
тестовой выборки \n", a)
print(classification_report(newsgroups_test.target, pred,
target_names=newsgroups_test.target_names))
clf = BernoulliNB(alpha=1)
for iin range(0, 100):
clf.fit(dense_test, newsgroups_test.target)
pred = clf.predict(dense_test)
```

[3] Данные «The 20 Newsgroups» — это коллекция примерно из 20000 новостных документов, разделенная (приблизительно) равномерно между 20 различными категориями. Коллекция «The 20 newsgroups» стала популярным набором данных для экспериментов с техниками машинного обучения для текстовых приложений, таких как классификация текста или его кластеризация.

```
# Точностьклассификации
a = metrics.accuracy_score(newsgroups_test.target, pred, normalize=True)
print("Точность классификации – доля верно классифицированных объектов из
тестовой выборки \n", a)
print(classification_report(newsgroups_test.target, pred,
target_names=newsgroups_test.target_names))
```

Результат работы программы:

```
Число наблюдений в обучающей выборке
 (11314,)
Число наблюдений в тестовой выборке
 (7532,)
Список новостных рубрик

['alt.atheism',
'comp.graphics',
 'comp.os.ms-windows.misc',
 'comp.sys.ibm.pc.hardware',
 'comp.sys.mac.hardware',
 'comp.windows.x',
 'misc.forsale',
 'rec.autos',
 'rec.motorcycles',
 'rec.sport.baseball',
 'rec.sport.hockey',
 'sci.crypt',
 'sci.electronics',
 'sci.med',
 'sci.space',
 'soc.religion.christian',
 'talk.politics.guns',
 'talk.politics.mideast',
 'talk.politics.misc',
 'talk.religion.misc']
['alt.atheism',
 'comp.graphics',
 'comp.os.ms-windows.misc',
 'comp.sys.ibm.pc.hardware',
 'comp.sys.mac.hardware',
 'comp.windows.x',
 'misc.forsale',
 'rec.autos',
 'rec.motorcycles',
 'rec.sport.baseball',
 'rec.sport.hockey',
 'sci.crypt',
 'sci.electronics',
 'sci.med',
 'sci.space',
 'soc.religion.christian',
 'talk.politics.guns',
 'talk.politics.mideast',
 'talk.politics.misc',
 'talk.religion.misc']
Размерность обучающей выборки sparse
 (11314, 101631)
Размерность тестовой выборки sparse
 (7532, 101631)
Размерность обучающей выборки dense
 (11314, 101631)
Размерность тестовой выборки dense
 (7532, 101631)
```

```
Точность классификации – доля верно классифицированных объектов из тестовой выборки
0.6355549654806161
                          precision    recall  f1-score   support

             alt.atheism       0.84      0.31      0.46       319
           comp.graphics       0.54      0.72      0.62       389
 comp.os.ms-windows.misc       0.20      0.00      0.01       394
comp.sys.ibm.pc.hardware       0.59      0.62      0.60       392
   comp.sys.mac.hardware       0.97      0.38      0.55       385
          comp.windows.x       0.39      0.89      0.54       395
            misc.forsale       0.93      0.59      0.72       390
               rec.autos       0.93      0.62      0.74       396
         rec.motorcycles       1.00      0.63      0.77       398
      rec.sport.baseball       1.00      0.63      0.78       397
        rec.sport.hockey       0.92      0.91      0.91       399
               sci.crypt       0.45      0.95      0.61       396
         sci.electronics       0.75      0.39      0.52       393
                 sci.med       0.64      0.84      0.73       396
               sci.space       0.73      0.86      0.79       394
  soc.religion.christian       0.49      0.93      0.64       398
      talk.politics.guns       0.69      0.66      0.68       364
   talk.politics.mideast       0.61      0.86      0.72       376
      talk.politics.misc       0.51      0.53      0.52       310
      talk.religion.misc       0.94      0.06      0.11       251

                accuracy                           0.64      7532
               macro avg       0.70      0.62      0.60      7532
            weighted avg       0.70      0.64      0.61      7532

Точность классификации – доля верно классифицированных объектов из тестовой выборки
0.780801911842804
                          precision    recall  f1-score   support

             alt.atheism       0.95      0.46      0.62       319
           comp.graphics       0.96      0.48      0.64       389
 comp.os.ms-windows.misc       0.89      0.85      0.87       394
comp.sys.ibm.pc.hardware       0.72      0.94      0.81       392
   comp.sys.mac.hardware       0.74      0.91      0.82       385
          comp.windows.x       0.94      0.86      0.90       395
            misc.forsale       0.37      0.98      0.54       390
               rec.autos       0.85      0.91      0.88       396
         rec.motorcycles       0.67      0.97      0.80       398
      rec.sport.baseball       0.87      0.89      0.88       397
        rec.sport.hockey       0.98      0.94      0.96       399
               sci.crypt       0.86      0.91      0.88       396
         sci.electronics       0.88      0.88      0.88       393
                 sci.med       0.90      0.74      0.82       396
               sci.space       0.94      0.87      0.90       394
  soc.religion.christian       0.73      0.82      0.77       398
      talk.politics.guns       0.80      0.76      0.78       364
   talk.politics.mideast       0.99      0.68      0.80       376
      talk.politics.misc       0.99      0.32      0.48       310
      talk.religion.misc       1.00      0.02      0.04       251

                accuracy                           0.78      7532
               macro avg       0.85      0.76      0.75      7532
            weighted avg       0.85      0.78      0.77      7532
```

5. Задания

Для набора данных «20 Newsgroups»

1. Подобрать оптимальное значение параметра α из интервала (0, 1)
2. Обучить классификатор с разными априорными вероятностями классов: равными и соответствующими долям классов в обучающей выборке