

# Анализ текстов

Данное задание основано на материалах лекций по методу опорных векторов.

## Вы научитесь:

- находить оптимальные параметры для метода опорных векторов
- работать с текстовыми данными

## Введение

Метод опорных векторов (Support Vector Machine, SVM) — один из видов линейных классификаторов. Функционал, который он оптимизирует, направлен на максимизацию ширины разделяющей полосы между классами. Из теории статистического обучения известно, что эта ширина тесно связана с обобщающей способностью алгоритма, а ее максимизация позволяет бороться с переобучением.

Одна из причин популярности линейных методов заключается в том, что они хорошо работают на разреженных данных. Так называются выборки с большим количеством признаков, где на каждом объекте большинство признаков равны нулю. Разреженные данные возникают, например, при работе с текстами. Дело в том, что текст удобно кодировать с помощью "мешка слов" — формируется столько признаков, сколько всего уникальных слов встречается в текстах, и значение каждого признака равно числу вхождений в документ соответствующего слова. Ясно, что общее число различных слов в наборе текстов может достигать десятков тысяч, и при этом лишь небольшая их часть будет встречаться в одном конкретном тексте.

Можно кодировать тексты хитрее, и записывать не количество вхождений слова в текст, а TF-IDF. Это показатель, который равен произведению двух чисел: TF (term frequency) и IDF (inverse document frequency). Первая равна отношению числа вхождений слова в документ к общей длине документа. Вторая величина зависит от того, в сколько документов выборки встречается это слово. Чем больше таких документов, тем меньше IDF. Таким образом, TF-IDF будет иметь высокое значение для тех слов, которые много раз встречаются в данном документе, и редко встречаются в остальных.

## Данные

Как мы уже говорили выше, линейные методы часто применяются для решения различных задач анализа текстов. В этом задании мы применим метод опорных векторов для определения того, к какой из тематик относится новость: атеизм или космос.

## Реализация в Scikit-Learn

Для начала вам потребуется загрузить данные. В этом задании мы воспользуемся одним из датасетов, доступных в scikit-learn'e — 20 newsgroups. Для этого нужно воспользоваться модулем datasets:

```
from sklearn import datasets

newsgroups = datasets.fetch_20newsgroups(
    subset='all',
    categories=['alt.atheism', 'sci.space']
)
```

После выполнения этого кода массив с текстами будет находиться в поле newsgroups.data, номер класса — в поле newsgroups.target.

Одна из сложностей работы с текстовыми данными состоит в том, что для них нужно построить числовое представление. Одним из способов нахождения такого представления является вычисление TF-IDF. В Scikit-Learn это реализовано в классе sklearn.feature\_extraction.text.TfidfVectorizer. Преобразование обучающей выборки нужно делать с помощью функции fit\_transform, тестовой — с помощью transform.

Реализация SVM-классификатора находится в классе `sklearn.svm.SVC`. Веса каждого признака у обученного классификатора хранятся в поле `coef_`.

Подбор параметров удобно делать с помощью класса `sklearn.grid_search.GridSearchCV` (При использовании библиотеки `scikit-learn` версии 18.0.1 и выше `sklearn.model_selection.GridSearchCV`). Пример использования:

```
grid = { 'C': np.power(10.0, np.arange(-5, 6)) }
cv = KFold(y.size, n_folds=5, shuffle=True, random_state=241)
clf = svm.SVC(kernel='linear', random_state=241)
gs = grid_search.GridSearchCV(clf, grid, scoring='accuracy', cv=cv)
gs.fit(X, y)
```

При использовании библиотеки `scikit-learn` версии 18.0.1 и выше `KFold` задаётся немного по-другому:

```
cv = KFold(n_splits=5, shuffle=True, random_state=241)
```

Первым аргументом в `GridSearchCV` передается классификатор, для которого будут подбираться значения параметров, вторым — словарь (`dict`), задающий сетку параметров для перебора. После того, как перебор окончен, можно проанализировать значения качества для всех значений параметров и выбрать наилучший вариант:

```
for a in gs.grid_scores_ :
    # a.mean_validation_score
    # a.parameters
```

## Инструкция по выполнению

1. Загрузите объекты из новостного датасета 20 newsgroups, относящиеся к категориям "космос" и "атеизм" (инструкция приведена выше).
2. Вычислите TF-IDF-признаки для всех текстов. Обратите внимание, что в этом задании мы предлагаем вам вычислить TF-IDF по всем данным. При таком подходе получается, что признаки на обучающем множестве используют информацию из тестовой выборки

— но такая ситуация вполне законна, поскольку мы не используем значения целевой переменной из теста. На практике нередко встречаются ситуации, когда признаки объектов тестовой выборки известны на момент обучения, и поэтому можно ими пользоваться при обучении алгоритма.

3. Подберите минимальный лучший параметр  $C$  из множества  $[10^{-5}, 10^{-4}, \dots, 10^4, 10^5]$  для SVM с линейным ядром (`kernel='linear'`) при помощи кросс-валидации по 5 блокам. Укажите параметр `random_state=241` и для SVM, и для KFold. В качестве меры качества используйте долю верных ответов (accuracy).
4. Обучите SVM по всей выборке с лучшим параметром  $C$ , найденным на предыдущем шаге.
5. Найдите 10 слов с наибольшим по модулю весом. Они являются ответом на это задание. Укажите их через запятую, в нижнем регистре, в лексикографическом порядке.

Ответ на каждое задание — текстовый файл, содержащий ответ в первой строчке. Обратите внимание, что отправляемые файлы не должны содержать перевод строки в конце. Данный нюанс является ограничением платформы Coursera. Мы работаем над тем, чтобы убрать это ограничение.