

Методы обработки данных. Задача классификации

Зуева Надежда
ФИВТ МФТИ

March 2018

1 Данные

- 1 Источники данных
- 2 Качество данных
- 3 Методы обработки

2 Выборки: train test

3 Функция ошибки

- 1 Что такое "ошибка"
- 2 Функционал ошибки
- 3 Accuracy
- 4 Оценка качества алгоритма

4 Классификация

- 1 Постановка задачи
- 2 Классификатор kNN
 - 1 Алгоритм
 - 2 Плюсы и минусы
 - 3 Модернизации kNN

X — множество **объектов**

Y — множество **допустимых ответов**

y^* — целевая функция, $y^*: X \rightarrow Y$, $y_i = y^*(x_i)$ известны только на **конечном** подмножестве объектов x_1, \dots, x_m из X

Пары (x_i, y_i) — прецеденты

Совокупность пар таких пар при i из $1, \dots, m$ — **обучающая выборка** (X_{train})

a — **решающая функция** (алгоритм), которая любому объекту из X ставит в соответствие допустимый ответ из Y и приближает целевую функцию y^*

X_{test} — **выборка прецедентов** для тестирования построенного алгоритма a

Для решения задачи обучения по прецедентам в первую очередь фиксируется восстанавливаемой зависимости.

Признак (feature) f объекта x — это результат измерения некоторой характеристики объекта. Формально признаком называется отображение $f : X \rightarrow D_f$, где D_f — множество допустимых значений признака. В частности, любой алгоритм $a : X \rightarrow Y$ также можно рассматривать как признак

Пусть дан набор признаков $f_1(x), \dots, f_n(x)$.







Признаковое описание объекта x — вектор (одномерный массив) (f_1, \dots, f_n) . Совокупность признаковых описаний всех объектов выборки длины m , записанную в виде таблицы размера mn , называют матрицей объектов–признаков.

Виды признаков

$$f : X \rightarrow D_f$$

- ❶ Бинарный признак: $D_f = 0, 1$
- ❷ Номинальный признак: $|D_f| < \infty$
- ❸ Порядковый признак: $|D_f| < \infty$ D_f — упорядочено
- ❹ Количественный признак: $D_f = \mathbb{R}$

Датасеты на www.kaggle.com

38		Brewer's Friend Beer Recipes Data on over 75,000 homemade beers jtrofe updated 2 days ago	food and drink alcohol chemistry	CSV 4 MB CC0	</> 6 2 4k
472		Data Science for Good: Kiva Crowdfunding Use Kernels to assess welfare of Kiva borrowers for \$30k in prizes Kiva updated 21 days ago	geography finance lending + 2 more...	CSV 42 MB CC0	</> 134 31 61k
14		ACLED African Conflicts, 1997-2017 Details on 165k Conflicts Across Africa Over Twenty Years Jacob Boysen updated 8 months ago	africa politics war	CSV 59 MB CC0	</> 3 1 2k
313		Huge Stock Market Dataset Historical daily prices and volumes of all U.S. stocks and ETFs Boris Marjanovic updated 4 months ago	business finance economics artificial intelligen...	Other 245 MB CC0	</> 13 8 36k
8		NOAA GOES-16 Next generation geostationary weather satellites data NOAA updated 10 days ago	earth sciences atmospheric scien... weather bigquery	BigQuery 9 GB CC0	</> 0 0 673
84		A Million News Headlines News headlines published over a period of 14 years. Rohk updated 3 months ago	news agencies historiography linguistics sociology	CSV 19 MB CC4	</> 26 4 17k

Как можно исследовать качество датасета?

```
df = pd.read_csv('assets/train.csv')  
df.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

- ❶ Достаточно данных для обучения
- ❷ Полнота (признаки)
- ❸ Полнота (объекты)
- ❹ Другие оценки

Датасеты в библиотеках

В библиотеке **Sklearn** есть набор датасетов, которые можно использовать. Например, датасет *iris*.

```
In [14]: from sklearn.datasets import load_iris
         data = load_iris()
         data.target[[10, 25, 50]]

         list(data.target_names)
```

```
Out[14]: ['setosa', 'versicolor', 'virginica']
```

```
In [16]: print(data)

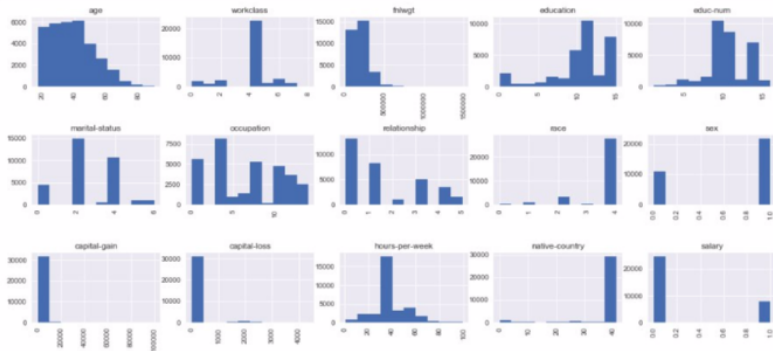
{'data': array([[5.1, 3.5, 1.4, 0.2],
                [4.9, 3. , 1.4, 0.2],
                [4.7, 3.2, 1.3, 0.2],
                [4.6, 3.1, 1.5, 0.2],
                [5. , 3.6, 1.4, 0.2],
                [5.4, 3.9, 1.7, 0.4],
                [4.6, 3.4, 1.4, 0.3],
                [5. , 3.4, 1.5, 0.2],
                [4.4, 2.9, 1.4, 0.2],
                [4.9, 3.1, 1.5, 0.1],
```

Подготовка датасета к работе с ним

- 1 Выкинуть дублирующие столбцы
- 2 Убрать пустые строки с помощью метода `fillna`
- 3 перекодировка категориальных признаков
- 4 графики!

Прогнозирование цен

```
fig = plt.figure(figsize=(19,8))
cols = 5
rows = np.ceil(float(encoded_data.shape[1]) / cols)
for i, column in enumerate(encoded_data.columns):
    ax = fig.add_subplot(rows, cols, i + 1)
    ax.set_title(column)
    encoded_data[column].hist(axes=ax)
    plt.xticks(rotation="vertical")
plt.subplots_adjust(hspace=0.7, wspace=0.2)
```



Обучающая выборка — выборка, по которой производится настройка (оптимизация параметров) модели зависимости.

Тестовая выборка — выборка, по которой оценивается качество построенной модели.

Функционал качества (обучение с учителем) — определяется как средняя ошибка ответов, выданных алгоритмом, по всем объектам выборки.

$x_i \in X, y_i \in Y$ $a(x_i)$ — наш алгоритм, y_i — верный ответ

- 1 доля верных ответов (она же — *accuracy*) :

$$R(a, X) = \frac{1}{|X|} \sum_{|X|} [a(x_i) == y_i]$$

- 2 доля ошибочных ответов: $W(a, X) = \frac{1}{|X|} \sum_{|X|} [a(x_i) \neq y_i]$

Качество алгоритма нельзя оценить по train sample!

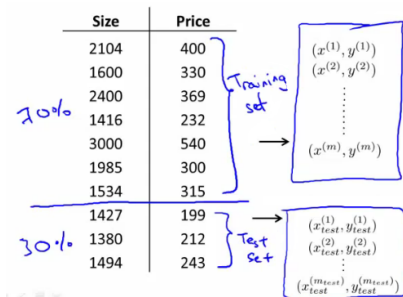
Разбиение выборки

Маленькая тестовая выборка, большая обучающая выборка

- (+) Обучающая выборка репрезентативная
- (−) Оценка качества ненадежная

Большая тестовая выборка, маленькая обучающая выборка

- (+) Оценка качества надежная
- (−) Оценка качества смещенная



Ирисы Фишера — это набор данных для задачи классификации, на примере которого Рональд Фишер в 1936 году продемонстрировал работу разработанного им метода дискриминантного анализа.



Ирис щетинистый
(*Iris setosa*)



Ирис разноцветный
(*Iris versicolor*)



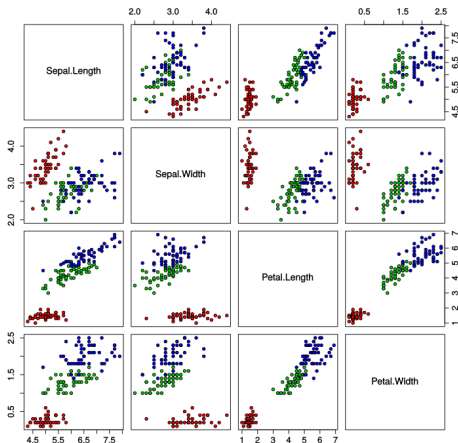
Ирис виргинский
(*Iris virginica*)

Расстояние

Близкие объекты обычно лежат в одном классе
какие объекты считать близкими?

Пусть $\rho(x, y)$ — функция расстояния

Iris Data (red=setosa, green=versicolor, blue=virginica)

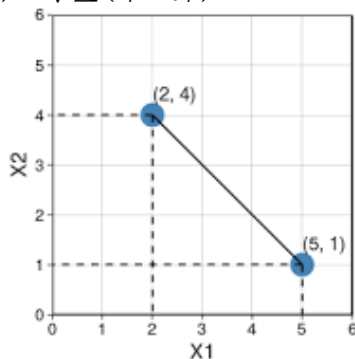


Измерение расстояния

Пусть $\rho(x, y)$ – функция расстояния

Евклидово расстояние:

$$\rho(x, y) = \sqrt{\sum (x_i^2 - y_i^2)}$$

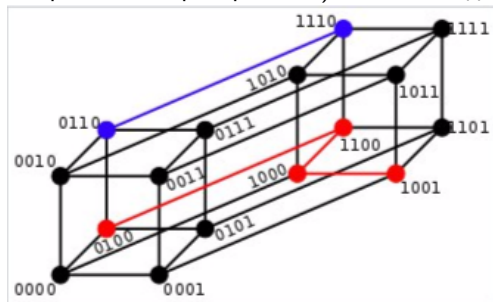


Измерение расстояния

Пусть $\rho(x, y)$ – функция расстояния

Расстояние Хэмминга:

число позиций, в которых соответствующие символы двух слов одинаковой длины различны. Расстояние Хэмминга применяется для строк одинаковой длины любых q -ичных алфавитов и служит метрикой различия (функцией, определяющей расстояние в метрическом пространстве) объектов одинаковой размерности.

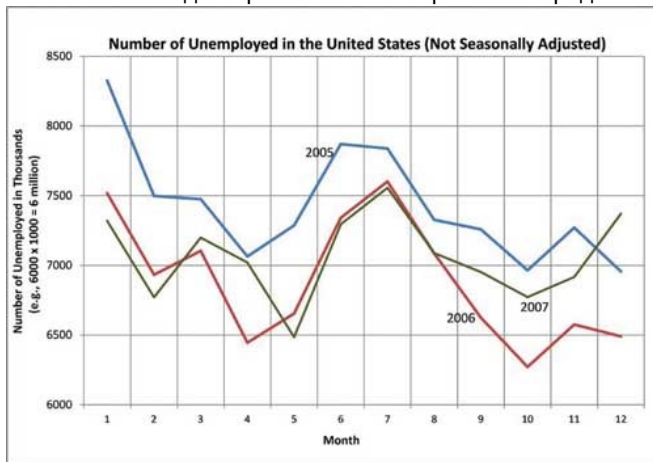


примеры расстояний в двоичном тессеракте:

расстояние 1 0110 → 1110, расстояние 3 0100 → 1001

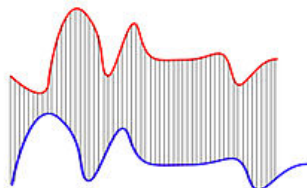
Измерение расстояния

Как можно вводить расстояние на временных рядах?

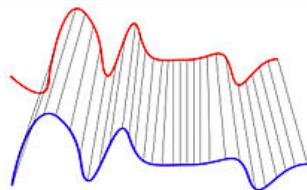


Измерение расстояния

Евклидово расстояние (пики, периоды)
DTWM



Euclidean Matching



Dynamic Time Warping Matching

Постановка задачи классификации

Имеется множество объектов, разделённых некоторым образом на классы.

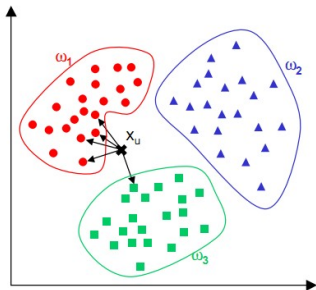
Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется **обучающей выборкой**. Классовая принадлежность остальных объектов не известна. Требуется построить **алгоритм** (классификатор), способный классифицировать произвольный объект из исходного множества.

Классифицировать объект — указать номер (или наименование класса), к которому относится данный объект. **Классификация объекта** — номер или *наименование класса*, выдаваемый **алгоритмом классификации** в результате его применения к данному конкретному объекту.

В математической статистике задачи классификации называются также задачами дискриминантного анализа.

Классификаторы.kNN

Метод ближайших соседей — метрический классификатор, основанный на оценивании *расстояний между объектами*. Классифицируемый объект относится к тому классу, которому принадлежат ближайшие к нему объекты обучающей выборки.

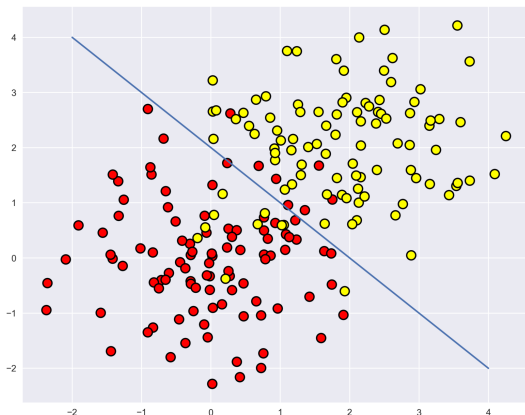


Для классификации каждого из объектов тестовой выборки необходимо **последовательно выполнить следующие операции**:

- 1 Вычислить расстояние до каждого из объектов обучающей выборки
- 2 Отобрать k объектов обучающей выборки, расстояние до которых минимально
- 3 Класс классифицируемого объекта — это класс, наиболее часто встречающийся среди k ближайших соседей

Выбор k

$k=0.6$



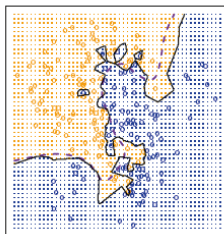
При выборе всех точек — вырождение в *const*

Алгоритм можно выразить формулой:

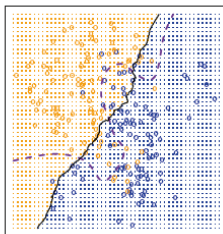
$$a(x) = \operatorname{argmax}(\sum_{i=1}^k a_i \cdot [y_{(i)} == y]), y \in Y$$
 Подбирается с помощью **holdout-выборки** или **кросс-валидации**

Чем больше k, тем проще разделяющая поверхность

KNN: K=1

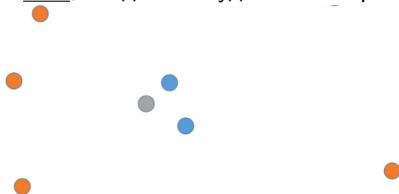


KNN: K=100



Проблема расстояний

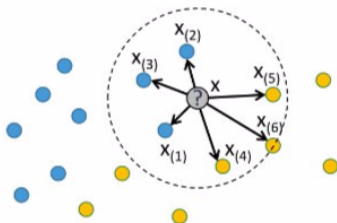
Пусть $k=5$, тогда как будет классифицирован объект?



Взвешенный kNN

Решение — учитывать расстояния среди k ближайших соседей: те объекты, которые расположены ближе, должны иметь больший вес.

Пример классификации ($k = 6$):





Home Installation Documentation Examples

Google Custom Search Search

Fort me on GitHub

Previous version 0.18.2 Next version 0.19.1 Up an Release

scikit-learn v0.19.1 Other versions

Please cite us if you use the software.

sklearn.neighbors.KNeighborsClassifier Examples using sklearn.neighbors.KNeighbors

sklearn.neighbors.KNeighborsClassifier

```
class sklearn.neighbors.KNeighborsClassifier(n_neighbors=5, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='minkowski', metric_params=None, n_jobs=1, **kwargs)
```

[\[source\]](#)

Classifier implementing the k-nearest neighbors vote.

Read more in the User Guide.

Parameters: `n_neighbors` : int, optional (default = 5)

Number of neighbors to use by default for `kneighbors` queries.

weights : str or callable, optional (default = 'uniform')

weight function used in prediction. Possible values:

- 'uniform' : uniform weights. All points in each neighborhood are weighted equally.
- 'distance' : weight points by the inverse of their distance. in this case, closer neighbors of a query point will have a greater influence than neighbors which are further away.
- [callable] : a user-defined function which accepts an array of distances, and returns an array of the same shape containing the weights.

algorithm : {'auto', 'ball_tree', 'kd_tree', 'brute'}, optional

Algorithm used to compute the nearest neighbors:

- 'ball_tree' will use [BallTree](#)