

# Введение в машинное обучение

Национальный исследовательский университет "Высшая школа экономики"  
Yandex School of Data Analysis

Неофициальный конспект по курсу.

1 апреля 2021 г.

## 1 Формальная постановка задачи машинного обучения

Машинное обучение последнее время применяется активно в прикладных задачах, где накоплены большие объемы информации и требуется решить такие задачи, как *предсказание, автоматизация принятий решений, классификация*, то есть такие задачи, которые ранее считались прерогативой человека.

Для решения подобных задач требуется:

1. Накопление большого количества данных;
2. Составление предсказательных моделей.

Именно вторым пунктом мы и будем заниматься в данном курсе.

### 1.1 Задача обучения по прецедентам

По сути дела, большая часть машинного обучения, это наука о том, как решать задачу *восстановления функции по точкам*.

Пусть у нас имеются данные, это множество объектов —  $X$ .

Также мы имеем множество ответов (размерность зависит от задачи) —  $Y$ .

Тогда наша предсказательная модель по своей сути — это отображение вида:

$y : X \rightarrow Y$  — неизвестная зависимость (**target function**)

Именно ее мы хотим найти, по крайней мере зная, что она промерена в конечном множестве точек, которое называется "*обучающей выборкой*":

$\{x_1, \dots, x_l\} \subset X$  — обучающая выборка (**training sample**)  
 $y_i = y(x_i), \quad i = 1, \dots, l$  — известные ответы.

То есть, грубо говоря, имеется  $l$  штук пар "объект-ответ" и нам хочется по этой информации восстановить эту зависимость, или построить функцию, которая будет аппроксимировать эту зависимость:

$a : X \rightarrow Y$  — алгоритм, решающий (аппроксимирующий) функцию (**decision function**), приближающую  $y$  на всем множестве  $X$ .

Таким образом, весь курс машинного обучения — это конкретизация:

- как задаются объекты и какими могут быть объекты;
- как строить функцию  $a$ ;
- в каком смысле  $a$  должен приближать  $y$ .

## 1.2 Как задаются объекты. Признаковое описание.

Самый распространенный способ задать описание объекта — признаковое описание.

Чисто формально, это функции, которые объектам ставят в соответствие какие-то значения (как правило числовые):

$$f_j : X \rightarrow D_j, \quad j = 1, \dots, n \text{ — признаки объектов (features)}$$

На содержательном уровне это какие-то способы измерения над объектами. В зависимости от того, что это за измерения, можно выделить различные типы признаков:

- $D_j = \{0, 1\}$  — бинарный признак  $f_j$ ;
- $|D_j| < \infty$  — номинальный признак  $f_j$ ;
- $|D_j| < \infty$ ,  $D_j$  упорядочено — порядковый признак  $f_j$ ;
- $D_j = \mathbb{R}$  — количественный признак  $f_j$ .

Для некоторого объекта  $x$  мы можем составить вектор с его описанием:

$$\text{Вектор } (f_1(x), \dots, f_n(x)) \text{ — признаковое описание объекта } x.$$

Важно, что количество различных значений для бинарного, номинального и порядкового **ограничено**. Для количественного признака, очевидно, это не справедливо.

Так же можно отметить, что часто для описания объекта используется *комбинация* различных типов признаков.

Главный объект, который далее будет применяться в решении нашей задачи, будет **матрица "объекты-признаки" (feature data)** как представление нашей обучающей выборки:

$$F = ||f_j(x_i)||_{l \times n} = \begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_l) & \dots & f_n(x_l) \end{pmatrix}$$

Строкам в этой матрице соответствуют признаковые описания соответствующего объекта (напомним, их  $l$  штук), а столбцам соответствуют признаки (конечное количество,  $n$  штук). Более того, каждой строке соответствует некоторый правильный ответ (очевидно, этого нет в матрице, но надо помнить, зачем она нам вообще нужна).

## 1.3 Как задаются ответы. Типы задач

### 1.3.1 Задачи классификации (classification)

- $Y = \{-1, +1\}$  — классификация на 2 класса.
- $Y = \{1, \dots, M\}$  — на  $M$  непересекающихся классов.

- $Y = \{0, 1\}^M$  — на  $M$  классов, которые могут пересекаться.

Классификация на 2 класса соответствует задачам, когда требуется принять одно из двух решений ("собака или кошка").

Второй вариант с  $M$  непересекающимися классами соответствует задачам с множественным выбором, например, оптическое распознавание символов рукописного текста.

Третий вариант с  $M$  пересекающимися классами подойдет, например, для медицинских задач — один больной имеет множество заболеваний.

### 1.3.2 Задачи кластеризации (cluster)

// формально на лекции не описывалось. Однако в целом, это задачи, похожие на классификацию, но группы, а иногда и их количество, заранее неизвестны.

### 1.3.3 Задачи восстановления регрессии (regression)

- $Y = \mathbb{R}$  или  $Y = \mathbb{R}^m$

Соответствует задачам, в которых ответы являются действительными числами. К этому классу задач принадлежат огромное количество задач прогнозирования, которые решаются в различных экономических, промышленных приложениях...

### 1.3.4 Задачи ранжирования (ranking, learning to rank)

- $Y$  — конечное упорядоченное множество.

Например, ранжирование запросов для поисковых сервисов.

## 1.4 Предсказательная модель

После постановки задачи мы двинемся далее, к вопросу о задании предсказательной модели.

Обычно **предсказательная модель (predictive model)** подбирается из некоторого семейства параметрических функций:

$$A = \{a(x) = g(x, \theta) \mid \theta \in \Theta\},$$

где  $g : X \times \Theta \rightarrow Y$  — фиксированная функция,

$\Theta$  — множество допустимых значений параметра  $\theta$ ,

$x$  — рассматриваемый объект.

Само семейство параметрических функций заранее изобретается, подбирается экспертами такое, что в нем нашлась бы функция, которая хорошо аппроксимирует нашу неизвестную зависимость. Обычно это некоторое семейство  $g(x, \theta)$ , где  $\theta$  — искомый вектор параметров нашей модели.

### 1.4.1 Линейная модель.

Самый простой и эффективный пример предсказательной модели — линейная модель.

*Линейная модель* с вектором параметров  $\theta = (\theta_1, \dots, \theta_n)$ ,  $\Theta = \mathbb{R}^n$ :

$$g(x, \theta) = \sum_{j=1}^n \theta_j \cdot f_j(x) \text{ — для регрессии и ранжирования, } Y = \mathbb{R}.$$

$$g(x, \theta) = \text{sign} \sum_{j=1}^n \theta_j \cdot f_j(x) \text{ — для классификации, } Y = \{-1, +1\}.$$

То есть для регрессии и ранжирования — это взвешенная сумма всех признаков (весами в данном случае являются значения вектора  $\theta$ ).

Для классификации нам просто достаточно знать, принадлежит ли объект к данному классу или нет (отсюда и использование  $\text{sign}$ ). То есть фактически, для задач классификации модель строит разделяющую гиперплоскость в  $n$ -мерном пространстве, по одну сторону которой находятся объекты одного класса, а по другую — другого.

И конечно возникает вопрос, почему сумма всех признаков взятых с некоторыми коэффициентами вообще является хорошей моделью для восстанавливаемой зависимости. Такой вопрос в целом возникает каждый раз при решении прикладных задач.

## 1.5 Этапы обучения и применения модели

Важно помнить, что для любой поставленной задачи машинного обучения, у нас всегда выделяются два этапа.

- **Этап обучения (train):**

Кратко: по выборке строим алгоритм-функцию, которая будет предсказывать нам какие-то значения на новых объектах.

Подробно: Метод обучения (*learning algorithm*)  $\mu : (X \times Y)^l \rightarrow A$  по выборке  $X^l = (x_i, y_i)_{i=1}^l$  строит алгоритм  $a = \mu(X^l)$ :

$$\left( \begin{array}{ccc} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_l) & \dots & f_n(x_l) \end{array} \right) \xrightarrow{y} \left( \begin{array}{c} y_1 \\ \dots \\ y_l \end{array} \right) \xrightarrow{\mu} a$$

- **Этап применения (test):**

Кратко: прогон алгоритма по новой выборке объектов.

Подробно: алгоритм  $a$  для новых объектов  $x'_1, \dots, x'_k$  выдает ответы  $a(x'_i)$ :

$$\left( \begin{array}{ccc} f_1(x'_1) & \dots & f_n(x'_1) \\ \dots & \dots & \dots \\ f_1(x'_k) & \dots & f_n(x'_k) \end{array} \right) \xrightarrow{a} \left( \begin{array}{c} a(x'_1) \\ \dots \\ a(x'_k) \end{array} \right)$$

**Возникает вопрос:** а каким образом мы будем выбирать из предсказательной модели тот алгоритм, который будет нас удовлетворять? Один из способов это сделать — **свести задачу к задаче оптимизации**. То есть выбрать такой алгоритм, который на большинстве объектов данной выборки будет точно или достаточно точно выдавать правильные ответы, но чтобы это сделать, нужно разобратся с тем, **как мы будем определять качество ответа для конкретного объекта**.

## 1.6 Функционалы качества

$\mathcal{L}(a, x)$  — функция потерь (**loss function**) — величина ошибки алгоритма  $a \in A$  на объекте  $x \in X$ .

**Функции потерь для задач классификации:**

- $\mathcal{L}(a, x) = [a(x) \neq y(x)]$  — индикатор ошибки;

**Функции потерь для задач регрессии:**

- $\mathcal{L}(a, x) = |a(x) - y(x)|$  — абсолютное значение ошибки;
- $\mathcal{L}(a, x) = (a(x) - y(x))^2$  — квадратичная ошибка.

*Эмпирический риск* — функционал качества алгоритма  $a$  на  $X^l$ :

$$Q(a, X^l) = \frac{1}{l} \sum_{i=1}^l \mathcal{L}(a, x_i)$$

И этот функционал качества уже можно минимизировать в задаче оптимизации.

*Кстати, не всегда берут среднее значение ошибки, то есть не всегда ставят  $\frac{1}{l}$ .*

## 1.7 Сведение задачи обучения к задаче оптимизации

*Минимизация эмпирического риска* **empirical risk minimization**:

$$\mu(X^l) = \arg \min_{a \in A} Q(a, X^l)$$

Как только мы сформулировали, что мы считаем ошибки и выписали функционал средней ошибки, мы можем уже решить нашу задачу оптимизации, применяя различные методы. Например, один из самых популярных — *метод наименьших квадратов* ( $Y = \mathbb{R}$ ,  $\mathcal{L}$  квадратична):

$$\mu(X^l) = \arg \min_{\theta} \sum_{i=1}^l (g(x_i, \theta) - y_i)^2$$

Метод удобен тем, что, во-первых, убирает знак, а во-вторых, в отличие от абсолютной разницы с модулем, удобно дифференцируем по параметрам.

## 1.8 Понятие обобщающей способности (generalization performance)

Теоретически все выглядит хорошо, но если искать подводные камни, можно задаться следующими вопросами:

- Найдём ли мы "закон природы" или *переобучимся*, то есть подгоним функцию  $g(x_i, \theta)$  под заданные точки?
- Будет ли  $a = \mu(X^l)$  приближать функцию  $y$  на всем  $X$ ?
- Будет ли  $Q(a, X^k)$  мало на новых данных — контрольной выборке  $X^k = (x'_i, y'_i)_{i=1}^k$ ,  $y'_i = y(x_i)$ ?

## 2 Резюме

**Основные понятия машинного обучения:**

- объект
- ответ

- признак
- предсказательная модель
- метод обучения
- эмпирический риск
- переобучение