Факультет вычислительной
математики и кибернетики

Как работает машинное обучение в искусственном интеллекте

А.В. Якушин, к.п.н., доцент, ВМК МГУ

Факультет ВМК МГУ имени М.В. Ломоносова
Неделя информационных технологий и кибербезопасности

Репозиторий с материалами

Что такое машинное обучение?

Машинное обучение

В одной из первых книг по машинному обучению «Machine Learning», написанной Томом Митчеллом в 1997, даётся следующее определение:

Говорят, что компьютерная программа учится из опыта $\mathcal{E}$ по отношению к какому-то классу задач $\mathcal{T}$ и критерию качества $\mathcal{P}$, если её качество на задачах из $\mathcal{T}$, измеренное с помощью $\mathcal{P}$, улучшается с использованием опыта $\mathcal{E}$.

Машинное обучение

Возникают следующие вопросы:

- Какие задачи решает машинное обучение? (класс задач $\mathcal{T}$) Почему эти задачи нельзя решить существующими методами?
- Что понимается под опытом в машинном обучении? (класс задач $\mathcal{E}$)
- Что такое критерий качества в машинном обучении? ($\mathcal{P}$)
- Что является результатом машинного обучения?

Какие задачи решает машинное обучение?

Какие задачи решает машинное обучение?

Машинное обучение это один из инструментов искусственного интеллекта, который решает задачи связанные с исследованием изучением реального мира, поиском закономерностей и научных открытий, создания изобретений, автоматизации промышленного производства и повседневной жизни.

Почему нужен новый способ решения задач?

Основной способ описания реальности в последние 300 лет это уравнения, в основном дифференциальные.

Дифференциальные уравнения описывают поведение некоторой системы в динамике, в зависимости от времени.

В природе система «хищник-жертва» взаимодействия популяций встречается достаточно часто. Например, в пруду обитают караси и щуки. В пруду достаточно питания карасям, а щуки питаются только карасями. По такой же системе взаимодействуют зайцы и волки, мыши и лисы и т. д.

$$\begin{cases} \frac{dx}{dt} = (\alpha - \gamma y)x; \\ \frac{dy}{dt} = (-\beta + \delta x)y. \end{cases}$$

где x — количество жертв, y — количество хищников, t — время, α,β,γ,δ — коэффициенты, отражающие взаимодействия между видами.

Почему нужен новый способ решения задач?

Мы перешли на новый уровень познания и аппарата дифференциальных уравнений уже не хватает для решения многих задач.

Почему?

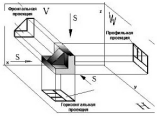
Потому что задачи стали сложные и зависят от огромного количества параметров.

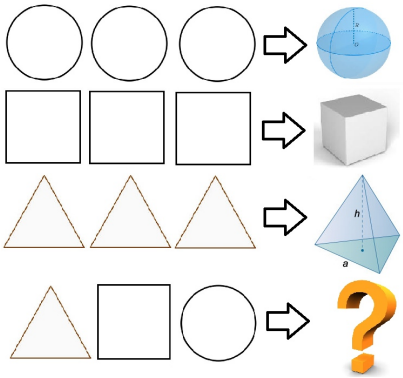
Статистика объема данных

- Каждый день в мире производится 2,5 квинтильона (10^{18}) байтов данных. 90% данных созданы за последние два года.
- Каждый час крупный интернет-магазин совершает 1 миллион сделок, пополняя базу данных на 2,5 петабайта (10^{15}) — в 170 раз больше объема данных Российской государственной библиотеки.
- Объем отправлений, доставляемых Почтой России службой за один год, равен 5 петабайтам, а Яндекс обрабатывает такой же объем данных всего за один час.
- Суммарный объем всей существующей на земле информации составляет несколько больше одного зеттабайта (10^{21}).

Необходимо собрать, организовать, проанализировать все эти данные.

И для этого нужны новые инструменты, такие как машинное обучение.





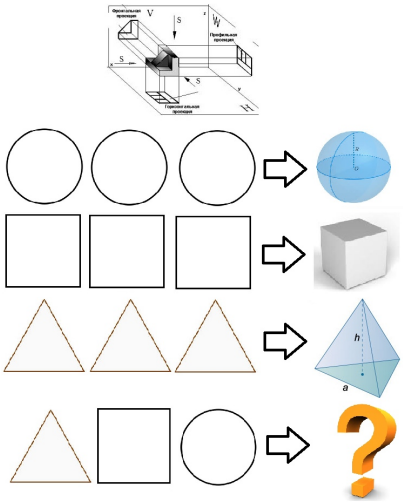
Классифицировать объект — значит, указать номер (или наименование класса), к которому относится данный объект.

Классификация объекта — номер или наименование класса, выдаваемый алгоритмом классификации в результате его применения к данному конкретному объекту.

Классификация бывает **бинарная** (два класса, да или нет, черный или белый и пр.) и **многоклассовая**.

Основные задачи анализа данных

Примеры задач классификации



1. Медицинская диагностика: по набору медицинских характеристик требуется поставить диагноз
2. Геологоразведка: по данным зондирования почв определить наличие полезных ископаемых
3. Оптическое распознавание текстов: по отсканированному изображению текста определить цепочку символов, его формирующих
4. Кредитный скоринг: по анкете заемщика принять решение о выдаче/отказе кредита
5. Синтез химических соединений: по параметрам химических элементов спрогнозировать свойства получаемого соединения

Один раджа приехал в гости к падишаху из соседнего государства и привез множество подарков, и среди них — живого слона. А так как падишах никогда в жизни не видел это животное, гость решил загадать загадку. Заведя слона в темный зал, раджа попросил правителя: «Пусть твои советники опишут слона, который здесь находится. Хочу узнать, насколько они мудры. Первый советник в кромешной тьме наткнулся на ногу животного. «Этот зверь, как громадное дерево», — сказал он. «Слон — это огромная извивающаяся змея», — возразил ему второй мудрец, который нащупал хобот животного. Третьему удалось погладить туловище слона. «О правитель, они оба лгут, — закричал он. — Слон похож на большой лист бумаги — такой же широкий и шершавый».

Недоумевающий падишах обратился к радже: «Скажи, каков этот слон?» Когда гость вывел животное из зала, и мудрецы, и правитель сильно удивились. «Каждый из вас был прав по-своему. И все же все вы ошибались, — обратился раджа к советникам. — Ваши знания были неполными, ведь вы познали лишь часть целого. А потому слон оказался совсем не таким, каким вы себе его представляли».

Что понимается под опытом в машинном обучении?

Под **опытом** в машинном обучении понимается специально подготовленный **набор данных**.

Данные

Дискретные или аналоговые сигналы, которые может обработать компьютер.

Знания

Большие массивы данных объединенные сложной внутренней структурой.

Неструктурированные данные

Данные неизвестной структуры классифицируются как **неструктурированные**. В дополнении к большим размерам, такая форма характеризуется рядом сложностей для обработки и извлечении полезной информации.

Типичный пример неструктурированных данных — источник, содержащий комбинацию простых текстовых файлов, картинок и видео. Сегодня организации имеют доступ к большому объему сырых или неструктурированных данных, но не знают как извлечь из них пользу.

Примером такой категории является результат Гугл поиска

Структурированные данные

Данные, которые могут храниться, быть доступными и обработанными в форме с фиксированным форматом называются **структурированными**. За продолжительное время компьютерные науки достигли больших успехов в совершенствовании техник для работы с этим типом данных (где формат известен заранее) и научились извлекать пользу.

Данные, хранящиеся в реляционной базе данных — структурированы и имеют вид, например, таблицы сотрудников компании.

Полуструктурированные данные

Эта категория содержит обе описанные выше, поэтому **полуструктурированные** данные обладают некоторой формой, но в действительности не определяются с помощью таблиц в реляционных базах. Пример этой категории — персональные данные, представленные в XML, CSV или JSON файле.

Пример XML

```
<?xml version="1.0"?>
<bibliography>
  <book> <!-- Информация по одной книге -->
    <title>Использование символа &amp;amp; в гипертекстах</title>
    <author>Иванов И. И.</author>
    <keywords><![CDATA[ символ, гипертекст, &]]> </keywords>
  </book>
  <book> <!-- Информация по одной книге -->
    <title>Вторая книга в списке</title>
    <author>Семенов И. И.</author>
    <keywords><![CDATA[ слово, событие, &]]> </keywords>
  </book>
</bibliography>
```

Пример CSV

```
Last_name;First_name;SSN;Test1;Test2;Test3;Test4;Final;Grade
"Alfalfa";"Aloysius";"123-45-6789";40.0;90.0;100.0;83.0;49.0;"D-"
"Alfred";"University";"123-12-1234";41.0;97.0;96.0;97.0;48.0;"D+"
"Gerty";"Gramma";"567-89-0123";41.0;80.0;60.0;40.0;44.0;"C"
"Android";"Electric";"087-65-4321";42.0;23.0;36.0;45.0;47.0;"B-"
"Bumpkin";"Fred";"456-78-9012";43.0;78.0;88.0;77.0;45.0;"A-"
"Rubble";"Betty";"234-56-7890";44.0;90.0;80.0;90.0;46.0;"C-"
"Noshow";"Cecil";"345-67-8901";45.0;11.0;-1.0;4.0;43.0;"F"
"Buff";"Bif";"632-79-9939";46.0;20.0;30.0;40.0;50.0;"B+"
"Airpump";"Andrew";"223-45-6789";49.01.0;90.0;100.0;83.0;"A"
"Backus";"Jim";"143-12-1234";48.0;1.0;97.0;96.0;97.0;"A+"
"Carnivore";"Art";"565-89-0123";44.0;1.0;80.0;60.0;40.0;"D+"
"Dandy";"Jim";"087-75-4321";47.0;1.0;23.0;36.0;45.0;"C+"
"Elephant";"Ima";"456-71-9012";45.0;1.0;78.0;88.0;77.0;"B-"
"Franklin";"Benny";"234-56-2890";50.0;1.0;90.0;80.0;90.0;"B-"
```

Пример JSON


```
{ "squadName": "Super hero squad",
  "homeTown": "Metro City",
  "formed": 2016,
  "secretBase": "Super tower",
  "members": [
    { "name": "Molecule Man",
      "age": 29,
      "secretIdentity": "Dan Jukes",
      "powers": ["Radiation resistance","Turning tiny","Radiation blast"]
    },
    { "name": "Madame Uppercut",
      "age": 39,
      "secretIdentity": "Jane Wilson",
      "powers": ["Million tonne punch","Damage resistance","Superhuman reflexes"]
    }
  ]
}
```

Пример RDF

Исходный документ:

Wells is the author of the book: The Invisible Man.

RDF документ:

```
<rdf>
  <Description about="https://www.xul.fr/Wells">
    <s:author>The Invisible Man</s:author>
  </Description>
</rdf>
```

Данные

Размеченные данные — это группа данных с присвоенными справочными тегами или выходной информацией. Например, массив фотографий котов, в котором указано, что это именно фотографии котов.

Данные о параметрах цветков ириса для определения вида ириса:

sepal_length	sepal_width	petal_length	petal_width	class(метка)
5.1	3.5	1.4	0.2	Iris-setosa
4.9	3.0	1.4	0.2	Iris-setosa
7.0	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
6.3	3.3	6.0	2.5	Iris-virginica
5.8	2.7	5.1	1.9	Iris-virginica

Данные

Неразмеченные данные — это группа данных, у которых отсутствует «метка».

Данные о клиентах магазина для проведения сегментации:

sex	age	marital	housing	balance
M	35	married	yes	1000
M	30	single	yes	2000
F	32	married	yes	500
M	32	married	no	300
F	33	single	no	30000
F	27	married	yes	4000

Типичный набор данных

Типичный набор данных содержит:

- Метаданные — описание полей данных их типы и назначение (семантика). Беъ этого набор данных имеет очень мало смысла.
- Данные — в одном из форматов XML, CSV, JSON.

Что является результатом машинного обучения?

Постановка задачи классификации

X — множество описаний объектов (признаки)

Y — истинная метка класса для каждого объекта (обычно метки обозначаются целыми числами: $\ldots -2, -1,0,1,2,3\ldots$

Наша задача найти зависимость между X и Y. То есть по имеющимся данным (обучающая выборка) найти функцию $f(X) = Y$, которая для любого объекта $x \in X$, способна поставить метку класса Y.

Будем рассматривать задачу бинарной классификации.

Классификация цветков ириса

Обратимся к классическому примеру классификации - классификации цветков ириса. Данные о цветках ириса стали, в некотором роде, стандартным набором данных для задачи классификации. Впервые были использованы в 1936 году (!) Рональдом Фишером для демонстрации работы разработанного им метода дискриминантного анализа - прародителя многих методов машинного обучения.

Требуется классифицировать цветки ириса на два вида:

- iris setosa
- iris virginica

Классификация цветков ириса

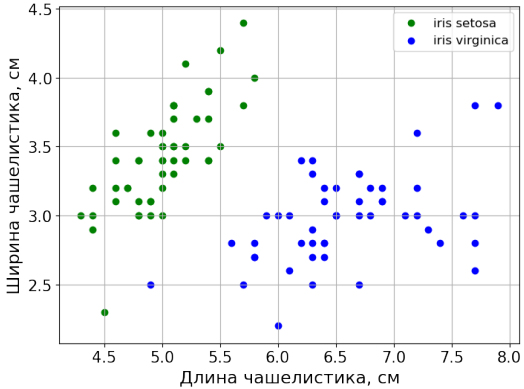
В качестве признаков рассмотрим два признака: длину и ширину чашелистика.

	Длина чашелистика	Ширина чашелистика	Вид Ириса
0	5.1	3.5	iris setosa
1	4.9	3.0	iris setosa
2	4.7	3.2	iris setosa
3	4.6	3.1	iris setosa
4	5.0	3.6	iris setosa
...	...	...	...
95	6.7	3.0	iris virginica
96	6.3	2.5	iris virginica
97	6.5	3.0	iris virginica
98	6.2	3.4	iris virginica
99	5.9	3.0	iris virginica

Первые 50 строчек таблицы соответсвуют виду iris setosa, остальные 50 iris virginica

Классификация цветков ириса

Визуализируем данные



Как видим точки на графике разбросаны не совсем хаотично, а имеют некоторую структуру, значит существует зависимость между параметрами чашелистика и видом цветка ириса.

Попробуем найти такую зависимость

Для предсказания меток класса по длине и ширине чашелистика, попробуем воспользоваться методом *линейная регрессия*:

$$\tilde{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2.$$

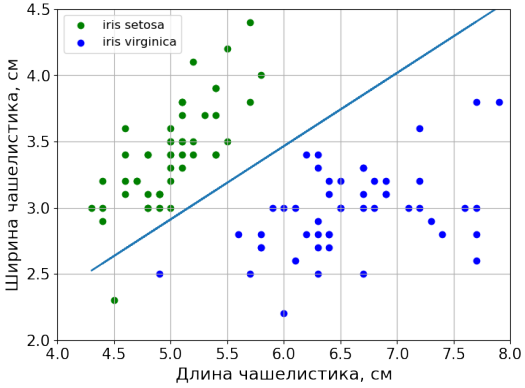
$\tilde{y}$ - ответ модели
x_1 - длина чашелистика
x_2 - ширина чашелистика
$\theta_0, \theta_1, \theta_2$ - параметры модели (веса)

Но как мы знаем, линейная регрессия выдает ответы в непрерывном диапазоне, а нам нужны ответы принимающие значения -1 или 1.

Для решения этой проблемы воспользуемся функцией знака от ответов линейной регрессии:

$$\tilde{y} = sign(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$$sign(t) = \begin{cases} 1, & t > 0 \\ 0, & t = 0 \\ -1, & t < 0 \end{cases}$$



Как видим, наши классы (виды цветка ириса) приблизительно можно разделить линией, значит имеет смысл построить такую разделяющую линию (или гиперплоскость так как ее размерность на 1 меньше размерности пространства).

В общем виде это уравнение записывается вот так:

$$\theta_1 x_1 + \theta_2 x_2 + b = 0$$

Или в более общем виде

$$X\Theta = 0,$$

где $X = (x_1, \ldots, x_n)$ и $\Theta = (\theta_1, \ldots, \theta_n)$.

Уравнение $X\Theta = 0$ задает гиперплоскость в пространстве признаков X с вектором весов Θ.

- Если $X_i\Theta > 0$, то i-й объект попал по одну сторону от гиперплоскости
- Если $X_i\Theta < 0$, то i-й объект попал по другую сторону от гиперплоскости.

А это значит, что наш *линейный классификатор*

$$\tilde{y}_i = sign(X_i\Theta),$$

будет давать метку
1, если объекты лежат по одну сторону от гиперплоскости
-1, если объекты лежать по другую сторону от гиперплоскости

Сам *линейный классификатор* строит **разделяющую гиперплоскость** в пространстве признаков.

Для поиска весов модели Θ, требуется задать функцию ошибки. Для задачи классификации, логичной функцией ошибки может служить **доля неправильных ответов** классификатора.

Введем следующие обозначения:
X - матрица признаков
y - истинные метки $X_i = \begin{pmatrix} 1 & x_{i,1} & x_{i,2} \end{pmatrix}$ - признаки i-го объекта (i-го цветка ириса)
$y_i \in \{$-1,1$\}$ - истинная метка на i-м объекте
$\Theta = \begin{pmatrix} \theta_0 & \theta_1 & \theta_2 \end{pmatrix}^T$
Введем понятие отступа на i-ом объекте: $M_i = y_i \cdot X_i\Theta$. Тогда $M_i > 0$, классификатор дал верный ответ и $M_i < 0$, классификатор ошибся

Тогда *долю неправильных ответов* легко посчитать следующим образом (напрямую такую функцию не получится использовать, но в рамках примера мы не будем углубляться в детали реализации):

$$L = \frac{1}{N}\sum_{i=1}^{N}[M_i < 0], \begin{cases} [True] = 1, \\ [False] = 0. \end{cases}$$

Задача машинного обучения для построения модели линейного классификатора (машинное обучение с учителем):

Необходимо найти такие коэффициенты θ_1 и θ_2, чтобы обеспечить минимальное значение функции ошибки L.

Модель машинного обучения $\tilde{y} = sign(0.05824189 + 0.78919385x_1 - 1.40657661x_2)$.

Применим модель к нашим данным (4 и 99 из таблицы на слайде 76)
$\tilde{y} = sign(-0.4482 + 1.2155 * 5.0 - 2.0299 * 3.6) = sign(-1.67834) = -1 =>$ iris setosa
$\tilde{y} = sign(-0.4482 + 1.2155 * 5.9 - 2.0299 * 3.0) = sign(0.63355) = 1 =>$ iris virginica

Выводы

Итак, перечислим основные этапы решения задачи машинного обучения:

1. Постановка задачи;
2. Выделение признаков;
3. Формирование выборки;
4. Выбор функционала ошибки;
5. Предобработка данных;
6. Построение модели;
7. Оценивание качества модели.

Выводы

Очевидно, что машинное обучение не следует применять везде, где требуется построить зависимость одной переменной от набора других.

Например, нет смысла восстанавливать зависимость силы от массы и ускорения — на эту задачу точный ответ даёт второй закон Ньютона.

Аналогично, нет смысла строить модель, которая выбирает сортирующую перестановку для массива чисел — это можно быстро и точно сделать с помощью, например, сортировки слиянием.