Введение в машинное обучение

Викулин Всеволод

v.vikulin@corp.mail.ru

30 сентября 2019

Часть 1

Введение в курс

Курс о машинном обучении

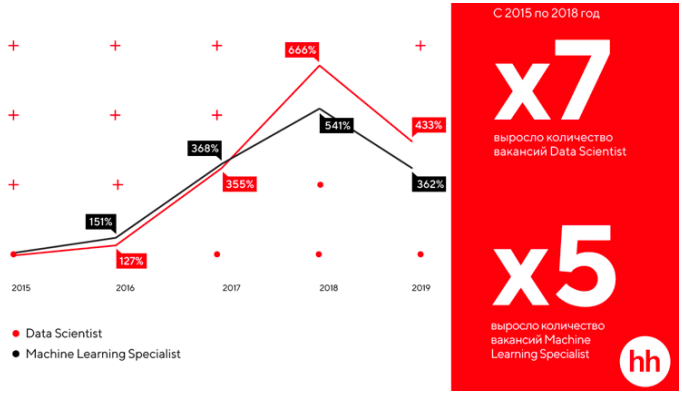
Машинное обучение (Machine Learning) — обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться
Искусственный интеллект (Artificial intelligence) — наука и технология создания интеллектуальных машин.
Источник: machinelearning.ru

Вопрос

Что значит способны обучаться? Кто их обучает?

Зачем все это?

- Специалисты по машинному обучению востребованы
- Много красивой математики
- Все применимо в реальных задачах



Знакомимся в с преподавателями

Всеволод Викулин

Дмитрий Меркушов

Дмитрий Парпулов

Сергей Чепарухин

Структура курса

- 11 лекций
- 2 коллоквиума (каждый 20 баллов)
- 4 домашних задания (первое 5 баллов, остальные 10, гибкая система штрафов)
- защита проекта (25 баллов)
- на каждой лекции небольшой тест по прошлой теме (10 бонусных баллов)
- море удовольствия (бесценно)

0–49 неудовлетворительно, 50–79 удовлетворительно, 80–94 хорошо, $>$ 94 отлично
Общаемся в слаке, домашние работы отправляем на **ml1.sphere@mail.ru**

Финальный проект

- Реальные данные от Mail.Ru
- Объединяемся в команды (максимум 4 человека)
- Решаем прикладную задачу на соревновательной платформе Kaggle
- Кто лучше решил, тот молодец
- Защищаем свое решение презентацией

Рекомендуемая литература

- Воронцов К.В. www.machinelearning.ru/wiki/images/6/6d/Voron-ML-1.pdf
- Bishop C. M. Pattern Recognition and Machine Learning
- Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning
- Skiena S. The Data Science Design Manual
- Ресурс www.machinelearning.ru
- Блог А.Г. Дьяконова www.dyakonov.org

Часть 2

Формальная постановка задачи машинного обучения

Общая постановка

Имеется множество объектов. Каждый объект описывается вектором его наблюдаемых характеристик (признаков) $x \in X$ и скрытых характеристик $y \in Y$ (целевая переменная). Существует некоторая функция $f : X \to Y$

Задача: имея **ограниченный** набор объектов (обучающая выборка), построить функцию $a : X \to Y$, приближающая f на всем множестве объектов (на генеральной совокупности).

Какие бывают признаки?

- Вещественный признак – принимает вещественные значения
- Бинарный признак – может принимать 2 значения
- Категориальный признак – может принимать K значений
- Порядковый признак – упорядоченный категориальный признак

Типы задач машинного обучения

Пусть обучающая выборка размера N. Обозначим:
$\{x_1, \ldots, x_N\} = X_{train}$, $\{y_1, \ldots, y_N\} = Y_{train}$

- Обучение с учителем (supervised learning). Известны X_{train}, Y_{train}
- Обучение без учителя (unsupervised learning). Известно только X_{train}
- Частичное обучение (semi-supervised learning). Известно X_{train} и для некоторых объектов из X_{train} известна целевая переменная

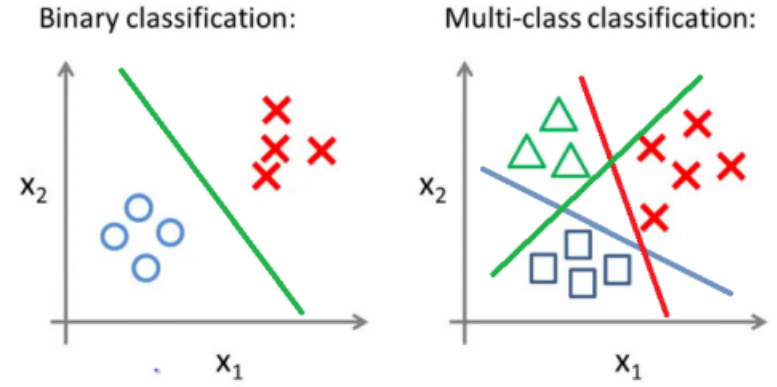
В нашем курсе рассмотрим первые два типа.

Обучение с учителем

По типу целевой переменной обучение с учителем тоже разбивается на классы.
В курсе разберем 2 постановки:

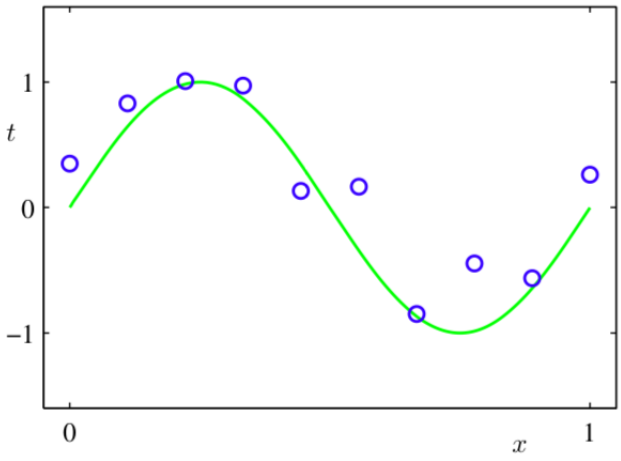
- Классификация – $Y = \{1, \dots, M\}$, классы могут пересекаться
- Регрессия – $Y = \mathbb{R}$ или $Y = \mathbb{R}^{\mathbb{M}}$

Пример классификации



Источник: medium.com/@b.terryjack

Пример регрессии



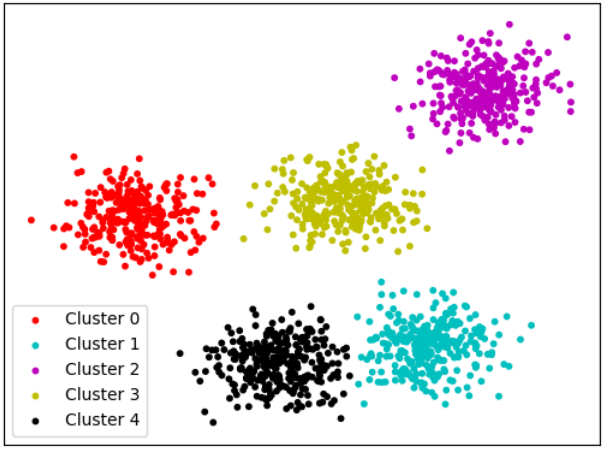
Источник: Bishop

Обучение без учителя

Можно выделить следующие типы:

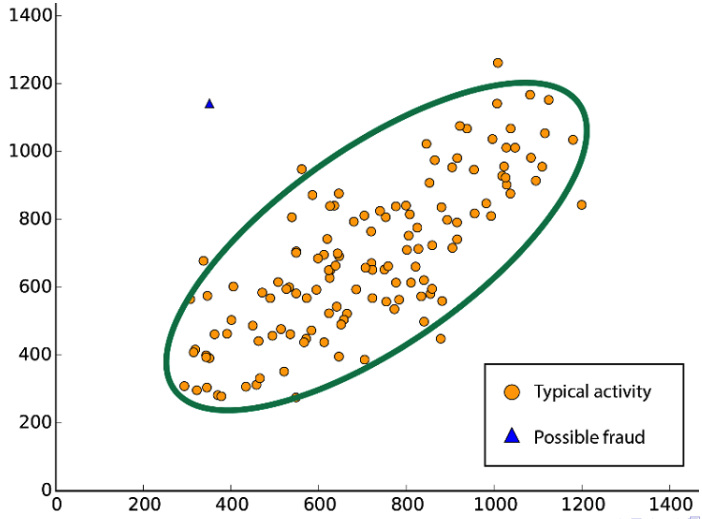
- Кластеризация – разбитие объектов на такие группы, что объекты в одних группах похожи, а в разных отличаются
- Поиск аномалий – поиск объектов, отличающихся от всех остальных
- Снижение размерности – уменьшение числа признаков

Пример кластеризации

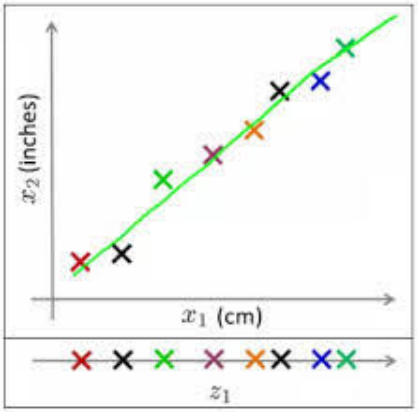


Источник: `towardsdatascience.com`

Пример поиска аномалий



Пример снижения размерности



Источник: analyticsvidhya.com/blog/2015/07/dimension-reduction-methods

Часть 3

Разбор прикладных задач

Правило разбора

Прежде чем делать прикладную задачу, нужно разобрать ее постановку!
Делаем по принципу:

1. Что является объектом в задаче?
2. Что является целевой переменной?
3. С учителем или без?
4. Регрессия или классификация? Кластеризация или поиск аномалий?
5. Какие данные нам нужны?
6. Какие признаки нужно извлечь?

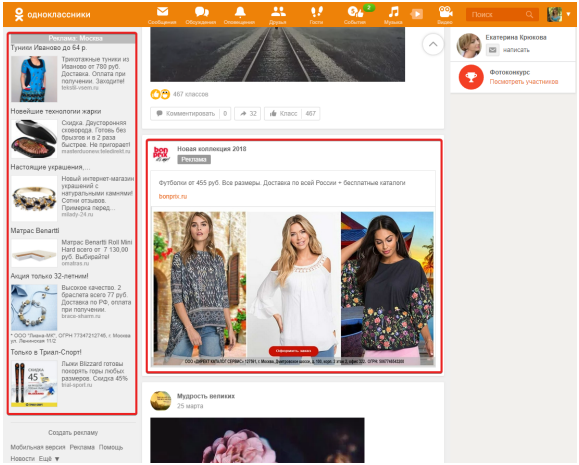
Спам-фильтр

Источник: technicallyeasy.net

Спам-фильтр

1. Письмо
2. Является ли письмо спамом
3. С учителем
4. Бинарная классификация
5. Письма, которые сами пользователи разметили, что это спам
6. Почта отправителя, содержит ли письмо маркерные фразы («скачать», «бесплатно», «без смс» и т.д.)

Рекламные объявления



Источник: edison.bz/blog/mytarget-sekrety-nastroek-v-2018-godu.html

Рекламные объявления

1. Пара (пользователь, объявление)
2. Кликнет ли пользователь на объявление
3. С учителем
4. Бинарная классификация
5. Пользовательская история взаимодействия с рекламой
6. Пол, возраст, город, интересы (интересуется ли он спортом, политикой и т.д.)

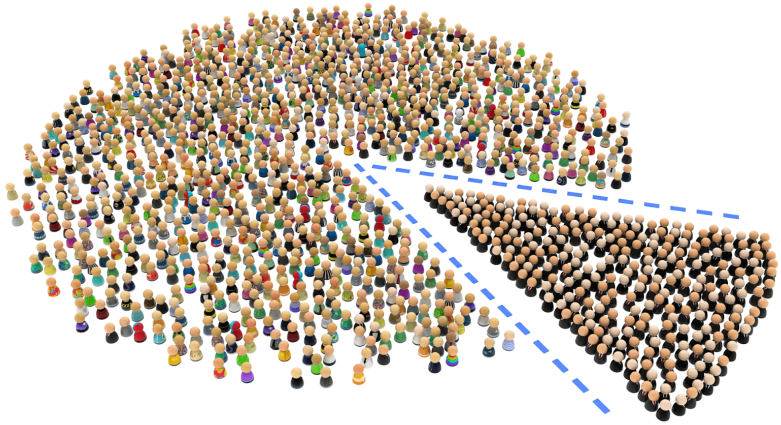
Предсказание объема продаж товара в магазине

Источник: assignmentpoint.com/business/finance/

Предсказание объема продаж товара в магазине

1. Тройка (товар, магазин, день)
2. Сколько мы за этот день продадим данного продукта в этом магазине?
3. С учителем
4. Регрессия
5. История продаж
6. Прошлые продажи, день недели, стоимость товара, есть ли скидка

Сегментация пользователей телеком компании



Источник: medium.com/analytics-for-humans

Сегментация пользователей телеком компании

1. Пользователь
2. Кластер пользователя
3. Без учителя
4. Кластеризация
5. История пользователя
6. Среднее время выхода пользователя в интернет, время звонков, сколько тратит в месяц, город

Детектирование поломок на заводе

Источник: strellagroup.com/industry-manufacturing-software

Детектирование поломок на заводе

1. Интервал времени
2. Есть ли поломка завода?
3. Без учителя
4. Поиск аномалий
5. История работы приборов на заводе
6. Отличие измерений от прошлых замеров на всех приборах

Часть 4

Как строить алгоритмы

Сравниваем алгоритмы

Далее рассматриваем обучение с учителем.

Функция потерь (loss function) $L(a, x, y)$ – неотрицательная функция, показывающая величину ошибки алгоритма a на объекте x с ответом y.

Функционал качества $Q(a, X, Y) = \frac{1}{N} \sum_{i=1}^{N} L(a, x_i, y_i), x_i \in X, y_i \in Y$

Принцип минимизации эмпирического риска:

$a^* = \underset{A}{\operatorname{argmin}} Q(a, X_{train}, Y_{train})$, где A – семейство алгоритмов.

Примеры функций потерь:

- Классификация – $L(a, x, y) = [a(x) = y]$
- Регрессия – $L(a, x, y) = |a(x) - y|$

Сравниваем алгоритмы

Самый важный вопрос: открыли ли мы закон природы или просто подогнали наш алгоритм $a(x)$ под обучающую выборку?
Не обязательно, что $\underset{A}{\operatorname{argmin}} Q(a, X_{train}, Y_{train})$ – полезный алгоритм.

Вопрос

Можете придумать пример алгоритма, у которого ошибка на обучении 0, но он совершенно бесполезен?

Финальный алгоритм проверяем на контрольной выборке X_{test}, Y_{test}, которую он раньше не видел.

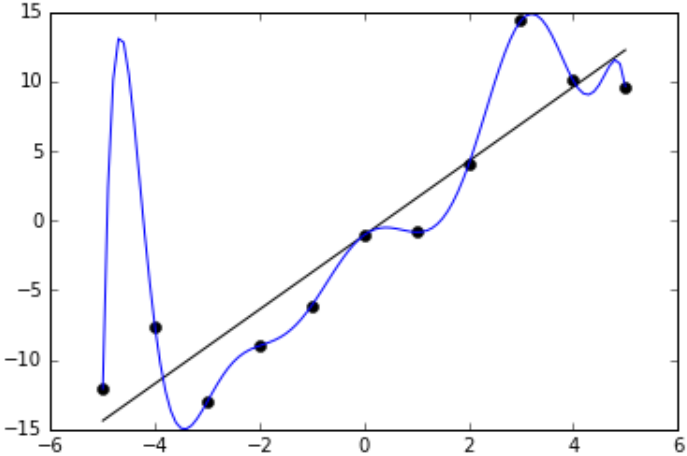
Вопрос

Как соотносятся $Q(a, X_{train}, Y_{train})$ и $Q(a, X_{test}, Y_{test})$

Переобучение и обобщающая способность

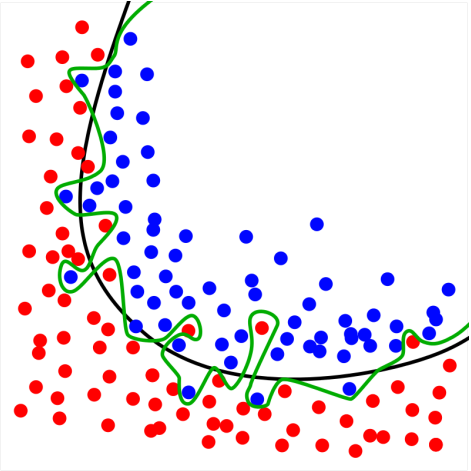
Проблема **переобучения** – значения $Q(a, X_{train}, Y_{train})$ значительно меньше, чем значение $Q(a, X_{test}, Y_{test})$ на контрольной выборке.
Если $Q(a, X_{test}, Y_{test})$ примерно равна $Q(a, X_{train}, Y_{train})$, то говорят, что алгоритм обладает **обобщающей способностью**
Переобучение есть всегда из-за индуктивной постановки задачи – нахождение закона природы по неполной выборке!
Но еще она может быть из-за излишней **сложности** модели.

Пример переобучения



Источник: en.wikipedia.org/wiki/Overfitting

Пример переобучения



Источник: en.wikipedia.org/wiki/Overfitting

Как обнаружить переобучение?

Было несколько подходов:

- Структурная минимизация риска (В. Вапник, А. Червоненкис, 1974)
- Информационный критерий Акаике (Акаике, 1974))
- Минимизация длины описания (Риссанен, 1978)
- Максимизация обоснованности (Маккай, 1992)

Надежно можно только эмпирически, посчитав разницу
$Q(a, X_{test}, Y_{test}) - Q(a, X_{train}, Y_{train})$

Как бороться с переобучением?

- Искать больше данных
- Упрощать семейство A, используя экспертные знания о структуре решения.

Важно

Без знания предметной области невозможно решать прикладную задачу. Нет идеального алгоритма, решающего все задачи лучше других.
The No Free Lunch Theorem, Wolpert, 1996

Заключение

Спасибо за внимание!