

Введение в искусственный интеллект. Машинное обучение

Лекция 1. Вводная. Кросс-валидация, дилемма смещения-разброса

Бабин Д.Н., Иванов И.Е., Петюшко А.А.

кафедра Математической Теории Интеллектуальных Систем

06 октября 2020 г.



① Организационные вопросы



План лекции

- 1 Организационные вопросы
- 2 Историческая справка



- 1 Организационные вопросы
- 2 Историческая справка
- 3 Постановка основных задач машинного обучения



- 1 Организационные вопросы
- 2 Историческая справка
- 3 Постановка основных задач машинного обучения
- 4 Тестирование моделей, кросс-валидация



- 1 Организационные вопросы
- 2 Историческая справка
- 3 Постановка основных задач машинного обучения
- 4 Тестирование моделей, кросс-валидация
- 5 Декомпозиция ошибки, недообучение и переобучение





Руководитель курса: д.ф.-м.н. Бабин Дмитрий Николаевич



Лектор: к.ф.-м.н. Иванов Илья Евгеньевич



Лектор: к.ф.-м.н. Петюшко Александр Александрович



- Авторы имеют более 15 лет опыта участия в проектах, связанных с машинным обучением и компьютерным зрением
- Являются постоянными участниками группы распознавания образов кафедры MaTIC
- В качестве научных консультантов работают или работали с такими крупнейшими российскими и международными компаниями как Нейроком, LSI Research, Fotonation, Huawei и др.



- В данный момент времени авторы ведут исследования в области компьютерного зрения в московском научно-исследовательском центре Хуавэй
- Данный курс является частью программы **SHARE**
 - **SHARE** = School of Huawei Advanced Research Education, или Школа опережающего научного образования Хуавэй
 - e-mail: share@intsys.msu.ru
 - Сайт SHARE: <http://sharemsu.ru>
 - Канал SHARE: https://t.me/joinchat/AAAAAE_r4XKzEDaUKy1FwA
 - Чат SHARE: <https://t.me/joinchat/AAAAAEnwHm0FStzFxKtS8w>



Зачем посещать этот курс

- 1 Специалисты по машинному обучению и анализу данных сейчас очень востребованы

Зачем посещать этот курс

- 1 Специалисты по машинному обучению и анализу данных сейчас очень востребованы
- 2 Шанс максимально использовать своё образование



Зачем посещать этот курс

- 1 Специалисты по машинному обучению и анализу данных сейчас очень востребованы
- 2 Шанс максимально использовать своё образование
- 3 Для лучших студентов возможны стажировки и бонусы от партнеров



Зачем посещать этот курс

- 1 Специалисты по машинному обучению и анализу данных сейчас очень востребованы
- 2 Шанс максимально использовать своё образование
- 3 Для лучших студентов возможны стажировки и бонусы от партнеров
- 4 И наконец, это просто интересно!



Что же такое искусственный интеллект?

Естественный интеллект (человек)

- Может воспринимать информацию, ее анализировать, принимать решения на основе анализа



Что же такое искусственный интеллект?

Естественный интеллект (человек)

- Может воспринимать информацию, ее анализировать, принимать решения на основе анализа

Искусственный интеллект

- (Сильный) то же самое, что и естественный, только на месте человека — компьютер



Что же такое искусственный интеллект?

Естественный интеллект (человек)

- Может воспринимать информацию, ее анализировать, принимать решения на основе анализа

Искусственный интеллект

- (Сильный) то же самое, что и естественный, только на месте человека — компьютер
- (Слабый) алгоритм, способный обучиться на основе массива входных данных, чтобы затем выполнять задачу вместо человека



Общая структура курса

«Введение в компьютерный интеллект»

1 Машинное обучение

- Необходимые основы для всего курса
- Предыдущие курсы лекций: [Весна 2019](#), [Весна 2020](#)



Общая структура курса

«Введение в компьютерный интеллект»

1 Машинное обучение

- Необходимые основы для всего курса
- Предыдущие курсы лекций: [Весна 2019](#), [Весна 2020](#)

2 Компьютерное зрение

- Извлечение информации из визуальных образов (изображений и видео)
- Предыдущий курс лекций: [Осень 2019](#)



Общая структура курса

«Введение в компьютерный интеллект»

1 Машинное обучение

- Необходимые основы для всего курса
- Предыдущие курсы лекций: [Весна 2019](#), [Весна 2020](#)

2 Компьютерное зрение

- Извлечение информации из визуальных образов (изображений и видео)
- Предыдущий курс лекций: [Осень 2019](#)

3 Обработка естественного языка

- Извлечение информации из речи и текста



Общая структура курса

«Введение в компьютерный интеллект»

1 Машинное обучение

- Необходимые основы для всего курса
- Предыдущие курсы лекций: [Весна 2019](#), [Весна 2020](#)

2 Компьютерное зрение

- Извлечение информации из визуальных образов (изображений и видео)
- Предыдущий курс лекций: [Осень 2019](#)

3 Обработка естественного языка

- Извлечение информации из речи и текста

4 Обучение с подкреплением

- Интерактивное взаимодействие со средой



- Предсказание стоимости недвижимости



Применение машинного обучения в реальной жизни

- Предсказание стоимости недвижимости
- Предсказание платёжеспособности клиента



Применение машинного обучения в реальной жизни

- Предсказание стоимости недвижимости
- Предсказание платёжеспособности клиента
- Предсказание оттока клиентов



Применение машинного обучения в реальной жизни

- Предсказание стоимости недвижимости
- Предсказание платёжеспособности клиента
- Предсказание оттока клиентов
- Классификация заболеваний



Применение машинного обучения в реальной жизни

- Предсказание стоимости недвижимости
- Предсказание платёжеспособности клиента
- Предсказание оттока клиентов
- Классификация заболеваний
- Предсказание клика пользователя по рекламному баннеру



Применение машинного обучения в реальной жизни

- Предсказание стоимости недвижимости
- Предсказание платёжеспособности клиента
- Предсказание оттока клиентов
- Классификация заболеваний
- Предсказание клика пользователя по рекламному баннеру
- И многие другие задачи...



Что будет в этом курсе

Теоретическая часть

- Постановка задач машинного обучения. Тестирование и метрики качества
 - Precision / Recall, TPR / FPR, ROC, AUC, Cross-Validation, ...
- Методы классификации и оптимизации
 - SVM, Random Forest, Decision Tree, Stochastic Gradient Descent, ...
- Методы восстановления регрессии
 - Linear Regression, Elastic Net, Ridge Regression, LASSO, ...
- Композиции алгоритмов
 - Bootstrapping, Bagging, Boosting, AdaBoost, GBoost, ...



Что будет в этом курсе

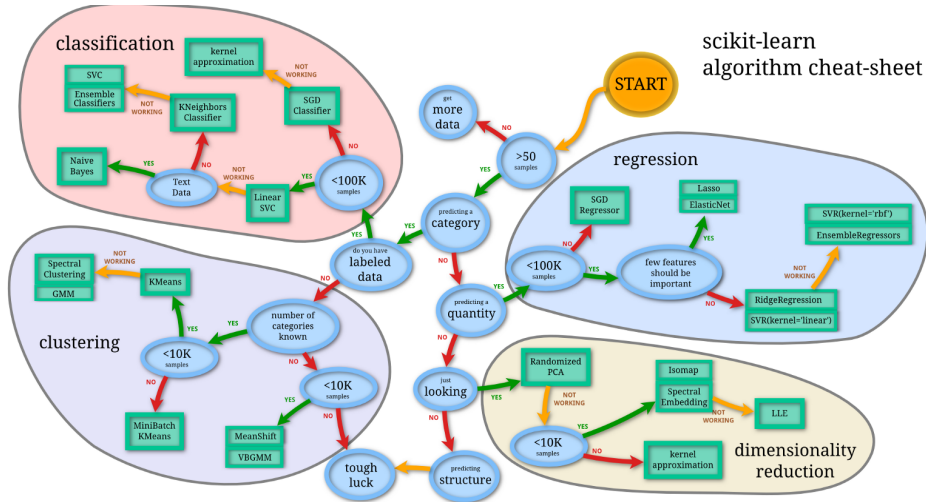
Теоретическая часть

- Постановка задач машинного обучения. Тестирование и метрики качества
 - Precision / Recall, TPR / FPR, ROC, AUC, Cross-Validation, ...
- Методы классификации и оптимизации
 - SVM, Random Forest, Decision Tree, Stochastic Gradient Descent, ...
- Методы восстановления регрессии
 - Linear Regression, Elastic Net, Ridge Regression, LASSO, ...
- Композиции алгоритмов
 - Bootstrapping, Bagging, Boosting, AdaBoost, GBoost, ...

Практическая часть

- Обработка и анализ данных на python
 - Scikit-Learn, Numpy, Pandas, ...
- Соревнования по машинному обучению

Дорожная карта Scikit-Learn¹



¹https://scikit-learn.org/stable/tutorial/machine_learning_map/

Чего не будет в этом курсе

- Глубокое обучение / Deep Learning

²См. курс “Математические основы цифровой обработки сигналов”, Мазуренко И. Л.

³См. курс “Математические основы цифровой обработки изображений”, Мазуренко И. Л.

Чего не будет в этом курсе

- Глубокое обучение / Deep Learning
- Частичное обучение / Semi-supervised Learning

²См. курс “Математические основы цифровой обработки сигналов”, Мазуренко И. Л.

³См. курс “Математические основы цифровой обработки изображений”, Мазуренко И. Л.

Чего не будет в этом курсе

- Глубокое обучение / Deep Learning
- Частичное обучение / Semi-supervised Learning
- Методы ранжирования / Ranking Systems

²См. курс “Математические основы цифровой обработки сигналов”, Мазуренко И. Л.

³См. курс “Математические основы цифровой обработки изображений”, Мазуренко И. Л.

Чего не будет в этом курсе

- Глубокое обучение / Deep Learning
- Частичное обучение / Semi-supervised Learning
- Методы ранжирования / Ranking Systems
- Прогнозирование временных рядов / Time Series Forecasting

²См. курс “Математические основы цифровой обработки сигналов”, Мазуренко И. Л.

³См. курс “Математические основы цифровой обработки изображений”, Мазуренко И. Л.

Чего не будет в этом курсе

- Глубокое обучение / Deep Learning
- Частичное обучение / Semi-supervised Learning
- Методы ранжирования / Ranking Systems
- Прогнозирование временных рядов / Time Series Forecasting
- Рекомендательные системы / Recommendation Systems

²См. курс “Математические основы цифровой обработки сигналов”, Мазуренко И. Л.

³См. курс “Математические основы цифровой обработки изображений”, Мазуренко И. Л.

Чего не будет в этом курсе

- Глубокое обучение / Deep Learning
- Частичное обучение / Semi-supervised Learning
- Методы ранжирования / Ranking Systems
- Прогнозирование временных рядов / Time Series Forecasting
- Рекомендательные системы / Recommendation Systems
- Цифровая обработка сигналов² и изображений³ / Digital Signal Processing and Digital Image Processing

²См. курс “Математические основы цифровой обработки сигналов”, Мазуренко И. Л.

³См. курс “Математические основы цифровой обработки изображений”, Мазуренко И. Л.

Оценки за курс

- Оценки за курс будут выставляться в соответствии с данными о посещении и набранными баллами за выполнение домашних заданий.



Оценки за курс

- Оценки за курс будут выставляться в соответствии с данными о посещении и набранными баллами за выполнение домашних заданий.
- В ходе курса будут предложены домашние задания трёх типов:
 - теоретические
 - практические
 - соревнования



Оценки за курс

- Оценки за курс будут выставляться в соответствии с данными о посещении и набранными баллами за выполнение домашних заданий.
- В ходе курса будут предложены домашние задания трёх типов:
 - теоретические
 - практические
 - соревнования
- В конце семестра состоится экзамен, на котором при желании можно будет повысить свою оценку



- Оценки за курс будут выставляться в соответствии с данными о посещении и набранными баллами за выполнение домашних заданий.
- В ходе курса будут предложены домашние задания трёх типов:
 - теоретические
 - практические
 - соревнования
- В конце семестра состоится экзамен, на котором при желании можно будет повысить свою оценку
- Предварительная шкала оценок:

Оценка	Процент выполненных заданий
Отлично	80 %
Хорошо	60 %
Зачет	40 %



- Оценки за курс будут выставляться в соответствии с данными о посещении и набранными баллами за выполнение домашних заданий.
- В ходе курса будут предложены домашние задания трёх типов:
 - теоретические
 - практические
 - соревнования
- В конце семестра состоится экзамен, на котором при желании можно будет повысить свою оценку
- Предварительная шкала оценок:

Оценка	Процент выполненных заданий
Отлично	80 %
Хорошо	60 %
Зачет	40 %

За посещение каждого занятия балл увеличивается примерно на 1%.



- Страница курса: <https://github.com/mlcoursemm/ml2020autumn>
- Главный ресурс по курсам “Введение в компьютерный интеллект”:
<https://github.com/mlcoursemm>
- Телеграмм-канал: <https://t.me/joinchat/AAAAAEUmx5cJL0dLXs0t8g>
- Группа обсуждения: <https://t.me/joinchat/B2iAGkx8IrWpQdM3LL-H-w>
- Почта курса: mlcoursemm@gmail.com
 - Именно сюда нужно будет посылать свои домашние задания!



Что такое машинное обучение

В 1959 году Артур Самуэль (Arthur Samuel) ввел в научный обиход термин “машинное обучение”.

Общее определение

Машинное обучение — процесс, в результате которого компьютеры способны показать поведение, которое в них не было явно запрограммировано.



Что такое машинное обучение

В 1959 году Артур Самуэль (Arthur Samuel) ввел в научный обиход термин “машинное обучение”.

Общее определение

Машинное обучение — процесс, в результате которого компьютеры способны показать поведение, которое в них не было явно запрограммировано.

В 1997 году Том М. Митчелл (Tom M. Mitchell) предложил формальное определение алгоритма машинного обучения.

Формальное определение

Считается, что **компьютерная программа обучается** по примерам E для некоторого набора задач T и метрики качества P , если качество ее работы на задачах из T , измеренное с помощью P , улучшается с использованием примеров E .



- Люди пытались предсказывать будущее на основе своего опыта с незапамятных времен.
- Однако научную основу заложили теория вероятностей (в особенности статистика) и линейная алгебра (как инструмент).



- Люди пытались предсказывать будущее на основе своего опыта с незапамятных времен.
- Однако научную основу заложили теория вероятностей (в особенности статистика) и линейная алгебра (как инструмент).
- **1795** г.: Гаусс впервые применяет метод наименьших квадратов (МНК) для анализа астрономических наблюдений. В **1805** г. Лежандр впервые публикует этот метод для анализа формы Земли. В настоящее время МНК является простейшим способом решить переопределенную систему линейных уравнений.



- Люди пытались предсказывать будущее на основе своего опыта с незапамятных времен.
- Однако научную основу заложили теория вероятностей (в особенности статистика) и линейная алгебра (как инструмент).
- **1795** г.: Гаусс впервые применяет метод наименьших квадратов (МНК) для анализа астрономических наблюдений. В **1805** г. Лежандр впервые публикует этот метод для анализа формы Земли. В настоящее время МНК является простейшим способом решить переопределенную систему линейных уравнений.
- **1901** г.: Карл Пирсон (Karl Pearson) изобрел метод главных компонент — главный метод уменьшения размерности данных.



- Люди пытались предсказывать будущее на основе своего опыта с незапамятных времен.
- Однако научную основу заложили теория вероятностей (в особенности статистика) и линейная алгебра (как инструмент).
- **1795** г.: Гаусс впервые применяет метод наименьших квадратов (МНК) для анализа астрономических наблюдений. В **1805** г. Лежандр впервые публикует этот метод для анализа формы Земли. В настоящее время МНК является простейшим способом решить переопределенную систему линейных уравнений.
- **1901** г.: Карл Пирсон (Karl Pearson) изобрел метод главных компонент — главный метод уменьшения размерности данных.
- **1906** г.: Андрей Андреевич Марков разрабатывает аппарат марковских цепей, который в **1913** г. применяет для исследования текста “Евгений Онегин”. Марковские цепи применяются для генерации и распознавания сигналов.



- **1950 г.:** Алан Тьюринг (Alan Turing) создает тест Тьюринга для оценки интеллекта компьютера.



- **1950** г.: Алан Тьюринг (Alan Turing) создает тест Тьюринга для оценки интеллекта компьютера.
- **1951** г.: Марвин Минский (Marvin Minsky) создал первую обучающуюся машину SNARC со случайно связанной нейросетью. В **1959** г. стал одним из сооснователей лаборатории искусственного интеллекта в MIT.



- **1950** г.: Алан Тьюринг (Alan Turing) создает тест Тьюринга для оценки интеллекта компьютера.
- **1951** г.: Марвин Минский (Marvin Minsky) создал первую обучающуюся машину SNARC со случайно связанной нейросетью. В **1959** г. стал одним из сооснователей лаборатории искусственного интеллекта в MIT.
- **1952** г.: Артур Самуэль создает первую шашечную программу для IBM 701. В **1955** г. Самуэль добавляет в программу способность к самообучению.



- **1950** г.: Алан Тьюринг (Alan Turing) создает тест Тьюринга для оценки интеллекта компьютера.
- **1951** г.: Марвин Минский (Marvin Minsky) создал первую обучающуюся машину SNARC со случайно связанной нейросетью. В **1959** г. стал одним из сооснователей лаборатории искусственного интеллекта в MIT.
- **1952** г.: Артур Самуэль создает первую шашечную программу для IBM 701. В **1955** г. Самуэль добавляет в программу способность к самообучению.
- **1958** г.: Фрэнк Розенблатт (Frank Rosenblatt) придумал Персептрон — первую искусственную нейронную сеть — и создал первый нейрокомпьютер “Марк-1”. *New York Times: Персептрон — это “эмбрион электронного компьютера, который в будущем сможет ходить, говорить, видеть, писать, воспроизводить себя и осознавать свое существование”.*



- **1950** г.: Алан Тьюринг (Alan Turing) создает тест Тьюринга для оценки интеллекта компьютера.
- **1951** г.: Марвин Минский (Marvin Minsky) создал первую обучающуюся машину SNARC со случайно связанной нейросетью. В **1959** г. стал одним из сооснователей лаборатории искусственного интеллекта в MIT.
- **1952** г.: Артур Самуэль создает первую шашечную программу для IBM 701. В **1955** г. Самуэль добавляет в программу способность к самообучению.
- **1958** г.: Фрэнк Розенблатт (Frank Rosenblatt) придумал Персептрон — первую искусственную нейронную сеть — и создал первый нейрокомпьютер “Марк-1”. *New York Times: Персептрон — это “эмбрион электронного компьютера, который в будущем сможет ходить, говорить, видеть, писать, воспроизводить себя и осознавать свое существование”.*
- **1963** г.: Лоуренс Робертс (Lawrence Roberts) сформулировал тезисы компьютерного зрения в своей диссертации в MIT.



- **1963** г.: Владимир Вапник и Алексей Червоненкис изобрели алгоритм SVM.



- **1963** г.: Владимир Вапник и Алексей Червоненкис изобрели алгоритм SVM.
- **1965** г.: Опубликована одна из первых книг по машинному обучению (классификации образов) — Nilsson N. Learning Machines, McGraw Hill.



- **1963** г.: Владимир Вапник и Алексей Червоненкис изобрели алгоритм SVM.
- **1965** г.: Опубликована одна из первых книг по машинному обучению (классификации образов) — Nilsson N. Learning Machines, McGraw Hill.
- **1966** г.: Джозеф Вейценбаум (Joseph Weizenbaum) написал виртуального собеседника ELIZA, способного имитировать (а скорее, пародировать) диалог с психотерапевтом (своим названием программа обязана героине из пьесы Б. Шоу).



- **1963** г.: Владимир Вапник и Алексей Червоненкис изобрели алгоритм SVM.
- **1965** г.: Опубликована одна из первых книг по машинному обучению (классификации образов) — Nilsson N. Learning Machines, McGraw Hill.
- **1966** г.: Джозеф Вейценбаум (Joseph Weizenbaum) написал виртуального собеседника ELIZA, способного имитировать (а скорее, пародировать) диалог с психотерапевтом (своим названием программа обязана героине из пьесы Б. Шоу).
- **1967** г.: Алексей Ивахненко и Валентин Лапа публикуют первый общий работающий обучающийся алгоритм для глубоких многослойных перцептронов для задач обучения с учителем.



- **1963** г.: Владимир Вапник и Алексей Червоненкис изобрели алгоритм SVM.
- **1965** г.: Опубликована одна из первых книг по машинному обучению (классификации образов) — Nilsson N. Learning Machines, McGraw Hill.
- **1966** г.: Джозеф Вейценбаум (Joseph Weizenbaum) написал виртуального собеседника ELIZA, способного имитировать (а скорее, пародировать) диалог с психотерапевтом (своим названием программа обязана героине из пьесы Б. Шоу).
- **1967** г.: Алексей Ивахненко и Валентин Лапа публикуют первый общий работающий обучающийся алгоритм для глубоких многослойных перцептронов для задач обучения с учителем.
- **1986** г.: Рина Дехтер (Rina Dechter) представила термин “Глубокое обучение” (Deep Learning) сообществу машинного обучения.



- **1963** г.: Владимир Вапник и Алексей Червоненкис изобрели алгоритм SVM.
- **1965** г.: Опубликована одна из первых книг по машинному обучению (классификации образов) — Nilsson N. Learning Machines, McGraw Hill.
- **1966** г.: Джозеф Вейценбаум (Joseph Weizenbaum) написал виртуального собеседника ELIZA, способного имитировать (а скорее, пародировать) диалог с психотерапевтом (своим названием программа обязана героине из пьесы Б. Шоу).
- **1967** г.: Алексей Ивахненко и Валентин Лапа публикуют первый общий работающий обучающийся алгоритм для глубоких многослойных перцептронов для задач обучения с учителем.
- **1986** г.: Рина Дехтер (Rina Dechter) представила термин “Глубокое обучение” (Deep Learning) сообществу машинного обучения.
- **1997** г.: Компьютер Deep Blue обыграл чемпиона мира по шахматам Гарри Каспарова.



- 2010 г.: Основание компании DeepMind.

⁴https://en.wikipedia.org/wiki/Perfect_information

- **2010** г.: Основание компании DeepMind.
- **2011** г.: Эндрю Ын (Andrew Ng), Грег Коррадо (Greg Corrado) и Джефф Дин (Jeff Dean) основали Google Brain.

⁴https://en.wikipedia.org/wiki/Perfect_information

- **2010** г.: Основание компании DeepMind.
- **2011** г.: Эндрю Ын (Andrew Ng), Грег Коррадо (Greg Corrado) и Джефф Дин (Jeff Dean) основали Google Brain.
- **2011** г.: Суперкомпьютер IBM Watson, оснащенный системой искусственного интеллекта, одержал победу в телевикторине *Jeopardy!*

⁴https://en.wikipedia.org/wiki/Perfect_information

- **2010** г.: Основание компании DeepMind.
- **2011** г.: Эндрю Ын (Andrew Ng), Грег Коррадо (Greg Corrado) и Джефф Дин (Jeff Dean) основали Google Brain.
- **2011** г.: Суперкомпьютер IBM Watson, оснащенный системой искусственного интеллекта, одержал победу в телевикторине *Jeopardy!*
- **2014** г.: В Facebook изобрели программный алгоритм DeepFace для распознавания лиц. Точность алгоритма составила 97%.

⁴https://en.wikipedia.org/wiki/Perfect_information

- **2010 г.:** Основание компании DeepMind.
- **2011 г.:** Эндрю Ын (Andrew Ng), Грег Коррадо (Greg Corrado) и Джефф Дин (Jeff Dean) основали Google Brain.
- **2011 г.:** Суперкомпьютер IBM Watson, оснащенный системой искусственного интеллекта, одержал победу в телевикторине *Jeopardy!*
- **2014 г.:** В Facebook изобрели программный алгоритм DeepFace для распознавания лиц. Точность алгоритма составила 97%.
- **2016 г.:** Программа AlphaGo, разработанная (уже) гугловской компанией DeepMind, выиграла в четырех играх из пяти у чемпиона мира по игре в го корейца Ли Седоля (Lee Se-dol). Компьютер выиграл в последней игре с полной информацией⁴ у человека (пример игры с неполной информацией — покер, хотя и там уже начинают успешно выступать роботы).

⁴https://en.wikipedia.org/wiki/Perfect_information

- **2010 г.:** Основание компании DeepMind.
- **2011 г.:** Эндрю Ын (Andrew Ng), Грег Коррадо (Greg Corrado) и Джефф Дин (Jeff Dean) основали Google Brain.
- **2011 г.:** Суперкомпьютер IBM Watson, оснащенный системой искусственного интеллекта, одержал победу в телевикторине *Jeopardy!*
- **2014 г.:** В Facebook изобрели программный алгоритм DeepFace для распознавания лиц. Точность алгоритма составила 97%.
- **2016 г.:** Программа AlphaGo, разработанная (уже) гугловской компанией DeepMind, выиграла в четырех играх из пяти у чемпиона мира по игре в го корейца Ли Седоля (Lee Se-dol). Компьютер выиграл в последней игре с полной информацией⁴ у человека (пример игры с неполной информацией — покер, хотя и там уже начинают успешно выступать роботы).
- **2016 г.:** При поддержке Илона Маска (Elon Musk) запущена некоммерческая исследовательская компания OpenAI.

⁴https://en.wikipedia.org/wiki/Perfect_information

Определения

- X — множество объектов
- Y — множество ответов
- $y : X \rightarrow Y$ — неизвестная зависимость



Способы машинного обучения

Определения

- X — множество объектов
- Y — множество ответов
- $y : X \rightarrow Y$ — неизвестная зависимость

Основные способы машинного обучения

- **С учителем** (сейчас)
 - Достаточное количество обучающего материала, то есть пар (x_i, y_i)
- Частичное обучение
 - Малое количество размеченных данных и много неразмеченных примеров x_i
- *Без учителя* (в следующих лекциях)
 - Нет размеченных пар, только примеры x_i
- С подкреплением
 - Формирование отклика на основе взаимодействия со средой

Постановка задачи обучения с учителем

- Дано:
 - $\{(x_1, y_1), \dots, (x_n, y_n)\} \subset X \times Y$ — обучающая выборка
- Найти
 - Решающую функцию $a : X \rightarrow Y$, которая приближает целевую зависимость y .



Постановка задачи обучения с учителем

- Дано:
 - $\{(x_1, y_1), \dots, (x_n, y_n)\} \subset X \times Y$ — обучающая выборка
- Найти
 - Решающую функцию $a : X \rightarrow Y$, которая приближает целевую зависимость y .
- Необходимо детализировать:
 - Как определяются объекты
 - Как задаются ответы
 - Что значит, что одна зависимость приближает другую



Как определяются объекты

Определение

Объект = совокупность признаков



Как определяются объекты

Определение

Объект = совокупность признаков

Типы признаков

- Категориальный признак



Как определяются объекты

Определение

Объект = совокупность признаков

Типы признаков

- Категориальный признак
- Бинарный признак
 - Частный случай категориального, когда категория = “есть данное свойство или нет”



Как определяются объекты

Определение

Объект = совокупность признаков

Типы признаков

- Категориальный признак
- Бинарный признак
 - Частный случай категориального, когда категория = “есть данное свойство или нет”
- Порядковый признак
 - Полный (частичный) порядок внутри категорий



Как определяются объекты

Определение

Объект = совокупность признаков

Типы признаков

- Категориальный признак
- Бинарный признак
 - Частный случай категориального, когда категория = “есть данное свойство или нет”
- Порядковый признак
 - Полный (частичный) порядок внутри категорий
- Количественный признак



Задачи классификации

- Бинарная классификация $Y = \{-1, 1\}$ или $Y = \{0, 1\}$
- Многоклассовая классификация $Y = \{0, 1, \dots, M - 1\}$
- Многозначная бинарная классификация $Y = \{0, 1\}^M$



Задачи классификации

- Бинарная классификация $Y = \{-1, 1\}$ или $Y = \{0, 1\}$
- Многоклассовая классификация $Y = \{0, 1, \dots, M - 1\}$
- Многозначная бинарная классификация $Y = \{0, 1\}^M$

Задачи восстановления регрессии

$Y = \mathbb{R}$ или $Y = \mathbb{R}^n$



Определение

Функция потерь (Loss function) $L(a, x)$ — величина ошибки алгоритма a на объекте x



Определение

Функция потерь (Loss function) $L(a, x)$ — величина ошибки алгоритма a на объекте x

Функции потерь для задач классификации

$L(a, x) = [a(x) \neq y]$ — индикатор ошибки



Функция потерь

Определение

Функция потерь (Loss function) $L(a, x)$ — величина ошибки алгоритма a на объекте x

Функции потерь для задач классификации

$L(a, x) = [a(x) \neq y]$ — индикатор ошибки

Функции потерь для задач восстановления регрессии

$L(a, x) = (a(x) - y)^2$ — квадратичная ошибка



Как понять, что одна модель лучше другой?

Для этого используют независимое от **обучающего** множества множество, которое называется **тестовым**



Как понять, что одна модель лучше другой?

Для этого используют независимое от **обучающего** множества множество, которое называется **тестовым**

Зачем вообще это понимать?

- Существует множество алгоритмов машинного обучения и важно понимать, какой из них более применим в конкретной задаче
- Даже в рамках одной модели может существовать множество параметров



Как выбирать лучшую модель

Наивный подход

Обучить модели с различными параметрами и выбрать лучшую на тесте



Как выбирать лучшую модель

Наивный подход

Обучить модели с различными параметрами и выбрать лучшую на тесте

Минусы наивного подхода

- Так как тест обычно состоит из случайной подвыборки исходной выборки, то результат на тесте тоже является некоторым приближением случайной величины
- Если все модели тестировать на тестовом датасете и таким образом выбирать лучшую, то будет происходить неявное обучение на тесте, а на другом независимом тесте возможны сюрпризы



Как выбирать лучшую модель

Наивный подход

Обучить модели с различными параметрами и выбрать лучшую на тесте

Минусы наивного подхода

- Так как тест обычно состоит из случайной подвыборки исходной выборки, то результат на тесте тоже является некоторым приближением случайной величины
- Если все модели тестировать на тестовом датасете и таким образом выбирать лучшую, то будет происходить неявное обучение на тесте, а на другом независимом тесте возможны сюрпризы

Что же делать?

Чтобы неявно не обучиться на тестовых данных — надо использовать кросс-валидацию, или скользящий контроль (cross-validation)

Общая идея

Основная идея кросс-валидации состоит в разбиении обучающего множества на два непересекающихся множества (возможно многократном):

$$X^{learn} = X^{train} \sqcup X^{val}$$

На одном из них происходит обучение, а на другом происходит валидация модели.



Общая идея

Основная идея кросс-валидации состоит в разбиении обучающего множества на два непересекающихся множества (возможно многократном):

$$X^{learn} = X^{train} \sqcup X^{val}$$

На одном из них происходит обучение, а на другом происходит валидация модели.

Зачем нужна валидация?

Обычно любой алгоритм машинного обучения содержит целый набор т.н. “**гиперпараметров**” (т.е. параметров, которые не обучаются, а задаются изначально): размерность, различные весовые коэффициенты и т.п. И для того, чтобы подбирать эти параметры “по-честному”, не используя вообще тестовые данные, и проводят процедуру валидации.

Частные случаи

- 1 Простейшая кросс-валидация — это контроль на отложенном множестве (hold-out), при котором происходит однократное разделение множества:

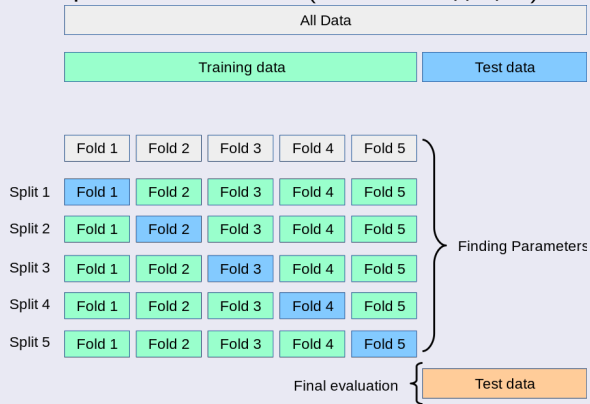
Train

Validation



Частные случаи

2 Контроль по k блокам (k-fold валидация)⁵:



⁵Image source: <https://scikit-learn.org/>

Частные случаи

- 3 Контроль по отдельным объектам (leave-one-out, или LOO валидация) — частный случай k -fold валидации, если k равно мощности обучающего множества



Частные случаи

- 3 Контроль по отдельным объектам (leave-one-out, или LOO валидация) — частный случай k -fold валидации, если k равно мощности обучающего множества
- 4 Многократная k -fold валидация — повторение k -fold валидации несколько раз с разными разбиениями.



Определение

Переобучение (overfitting) — нежелательное явление, возникающее при решении задач обучения по прецедентам, когда вероятность ошибки обученного алгоритма на объектах тестовой выборки оказывается существенно выше, чем средняя ошибка на обучающей выборке.

Переобучение возникает при использовании избыточно сложной модели



Определение

Переобучение (overfitting) — нежелательное явление, возникающее при решении задач обучения по прецедентам, когда вероятность ошибки обученного алгоритма на объектах тестовой выборки оказывается существенно выше, чем средняя ошибка на обучающей выборке.

Переобучение возникает при использовании избыточно сложной модели

Одна из основных причин возникновения

Избыточная размерность пространства параметров модели, “лишние” степени свободы используются для точной настройки на обучающую выборку



Переобучение

Определение

Переобучение (overfitting) — нежелательное явление, возникающее при решении задач обучения по прецедентам, когда вероятность ошибки обученного алгоритма на объектах тестовой выборки оказывается существенно выше, чем средняя ошибка на обучающей выборке.

Переобучение возникает при использовании избыточно сложной модели

Одна из основных причин возникновения

Избыточная размерность пространства параметров модели, “лишние” степени свободы используются для точной настройки на обучающую выборку

Один из основных методов обнаружения

Использование кросс-валидации

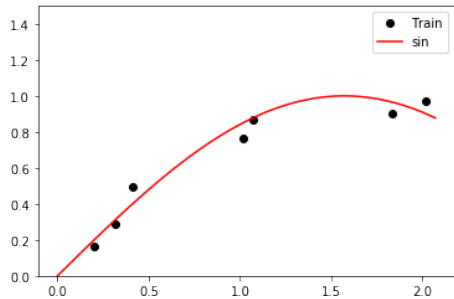
Определение

Недообучение (underfitting) – нежелательное явление, возникающее при решении задач обучения по прецедентам, когда алгоритм обучения не обеспечивает достаточно малой величины средней ошибки на обучающей выборке.

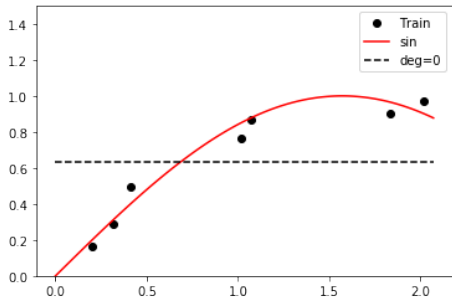
Недообучение возникает при использовании недостаточно сложных моделей



Примеры недообучения и переобучения



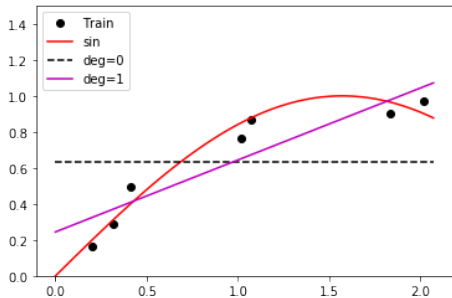
Примеры недообучения и переобучения



- Полином нулевой степени не может хорошо приближать зависимость в силу ограниченности параметров модели



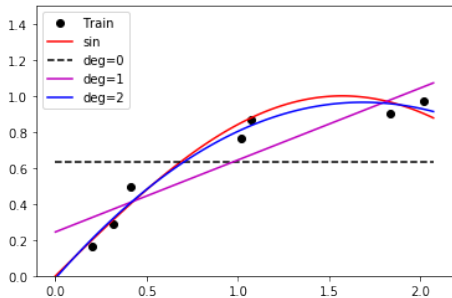
Примеры недообучения и переобучения



- Полином нулевой степени не может хорошо приближать зависимость в силу ограниченности параметров модели
- Линейная и квадратичная модели адекватно описывают закономерность



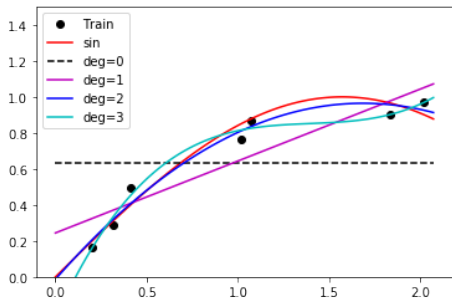
Примеры недообучения и переобучения



- Полином нулевой степени не может хорошо приближать зависимость в силу ограниченности параметров модели
- Линейная и квадратичная модели адекватно описывают закономерность
- Полиномы высоких степеней могут в точности пройти через точки обучающей выборки



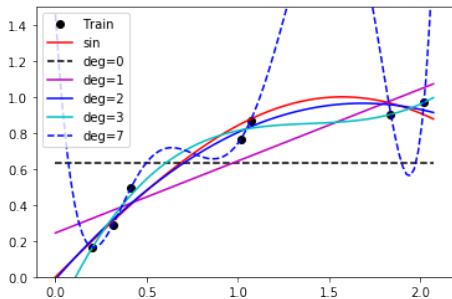
Примеры недообучения и переобучения



- Полином нулевой степени не может хорошо приближать зависимость в силу ограниченности параметров модели
- Линейная и квадратичная модели адекватно описывают закономерность
- Полиномы высоких степеней могут в точности пройти через точки обучающей выборки



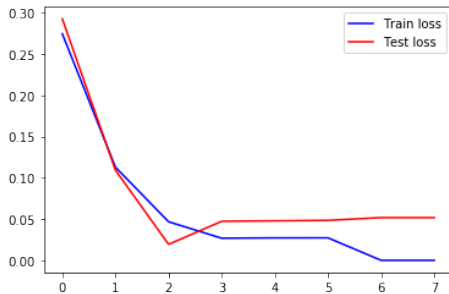
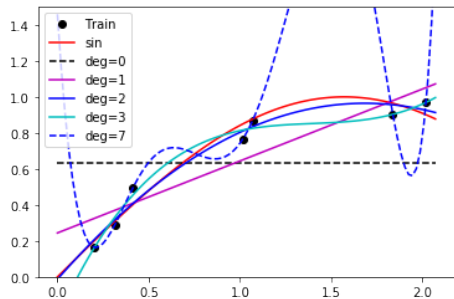
Примеры недообучения и переобучения



- Полином нулевой степени не может хорошо приближать зависимость в силу ограниченности параметров модели
- Линейная и квадратичная модели адекватно описывают закономерность
- Полиномы высоких степеней могут в точности пройти через точки обучающей выборки



Примеры недообучения и переобучения



- Полином нулевой степени не может хорошо приближать зависимость в силу ограниченности параметров модели
- Линейная и квадратичная модели адекватно описывают закономерность
- Полиномы высоких степеней могут в точности пройти через точки обучающей выборки



О параметрах и гиперпараметрах

В примере с приближением неизвестной зависимости полиномом

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0:$$

- **Параметры:** коэффициенты $a_n, a_{n-1}, \dots, a_1, a_0$, и они настраиваются во время обучения модели
- **Гиперпараметры:** степень многочлена n , которая выбирается до начала обучения и затем выбирается из множества проверенных на валидационном множестве



Вывод выражения среднеквадратичной ошибки

Определения

Пусть $y = y(x) = f(x) + \varepsilon$ — целевая зависимость, где $f(x)$ — детерминированная функция, $\varepsilon \sim N(0, \sigma^2)$ и $a(x)$ — алгоритм машинного обучения.



Вывод выражения среднеквадратичной ошибки

Определения

Пусть $y = y(x) = f(x) + \varepsilon$ — целевая зависимость, где $f(x)$ — детерминированная функция, $\varepsilon \sim N(0, \sigma^2)$ и $a(x)$ — алгоритм машинного обучения.

Полагаем, что ε и a — независимые ($Ea\varepsilon = EaE\varepsilon$). $Ey = Ef$, $Dy = D\varepsilon = \sigma^2$.

Разложение квадрата ошибки

$$\begin{aligned} E(y - a)^2 &= E(y^2 + a^2 - 2ya) = Ey^2 + Ea^2 - 2Eya = \\ &= Ey^2 + Ea^2 - 2E(f + \varepsilon)a = Ey^2 + Ea^2 - 2Efa - 2E\varepsilon a = \\ &= Ey^2 - (Ey)^2 + (Ey)^2 + Ea^2 - (Ea)^2 + (Ea)^2 - 2fEa = \\ &= Dy + Da + (Ey)^2 + (Ea)^2 - 2fEa = Dy + Da + (Ef)^2 - 2fEa + (Ea)^2 = \\ &= Dy + Da + (E(f - a))^2 = \sigma^2 + \text{variance}(a) + \text{bias}^2(f, a) \end{aligned}$$

Определение

Разброс (variance) — дисперсия ответов алгоритмов $a(x)$.

Характеризует разнообразие алгоритмов (из-за случайности обучающей выборки, шума, стохастичности обучения и т.д.)



Дополнительные определения

Определение

Разброс (variance) — дисперсия ответов алгоритмов $a(x)$.

Характеризует разнообразие алгоритмов (из-за случайности обучающей выборки, шума, стохастичности обучения и т.д.)

Определение

Смещение (bias) — матожидание разности между истинным ответом и выбранным алгоритмом.

В примере выше — это $E(f - a)$.

Характеризует способность модели настраиваться на целевую зависимость



Дополнительные определения

Определение

Разброс (variance) — дисперсия ответов алгоритмов $a(x)$.

Характеризует разнообразие алгоритмов (из-за случайности обучающей выборки, шума, стохастичности обучения и т.д.)

Определение

Смещение (bias) — матожидание разности между истинным ответом и выбранным алгоритмом.

В примере выше — это $E(f - a)$.

Характеризует способность модели настраиваться на целевую зависимость

Определение

Разложение квадрата ошибки в примере выше называется **дилеммой смещения-разброса** (bias-variance tradeoff)

Модель оптимальной сложности: классический взгляд

- Для простых моделей характерно недообучение



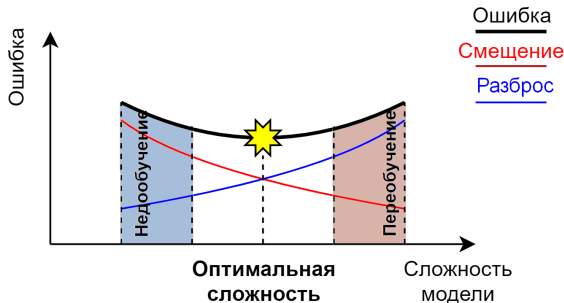
Модель оптимальной сложности: классический взгляд

- Для простых моделей характерно недообучение
- Для сложных моделей характерно переобучение



Модель оптимальной сложности: классический взгляд

- Для простых моделей характерно недообучение
- Для сложных моделей характерно переобучение
- Оптимальная сложность модели где-то между



Модель оптимальной сложности: современные эмпирические данные

- Ранее не было технической возможности посмотреть на качество работы в случае модели огромной сложности

⁶Advani, Madhu S., Andrew M. Saxe, and Haim Sompolsky. "High-dimensional dynamics of generalization error in neural networks." 2017

Модель оптимальной сложности: современные эмпирические данные

- Ранее не было технической возможности посмотреть на качество работы в случае модели огромной сложности
- С развитием техники стало возможным обучать модели с миллионами и даже миллиардами параметров

⁶Advani, Madhu S., Andrew M. Saxe, and Haim Sompolinsky. "High-dimensional dynamics of generalization error in neural networks." 2017

Модель оптимальной сложности: современные эмпирические данные

- Ранее не было технической возможности посмотреть на качество работы в случае модели огромной сложности
- С развитием техники стало возможным обучать модели с миллионами и даже миллиардами параметров
- Оказалось, что с увеличением сложности ошибка ведет себя сначала как предсказывает дилемма смещения-разброса, а затем неожиданно начинает снижаться⁶ и выходит даже на меньший уровень ошибки!

⁶Advani, Madhu S., Andrew M. Saxe, and Haim Sompolinsky. "High-dimensional dynamics of generalization error in neural networks." 2017

Модель оптимальной сложности: современные эмпирические данные

- Ранее не было технической возможности посмотреть на качество работы в случае модели огромной сложности
- С развитием техники стало возможным обучать модели с миллионами и даже миллиардами параметров
- Оказалось, что с увеличением сложности ошибка ведет себя сначала как предсказывает дилемма смещения-разброса, а затем неожиданно начинает снижаться⁶ и выходит даже на меньший уровень ошибки!
- Переломный момент — точка, в которой сложность модели сопоставима с мощностью обучающей выборки (interpolation threshold)

⁶Advani, Madhu S., Andrew M. Saxe, and Haim Sompolinsky. “High-dimensional dynamics of generalization error in neural networks.” 2017

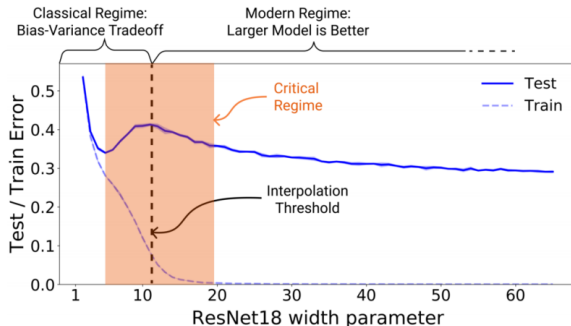
Модель оптимальной сложности: современные эмпирические данные

- Ранее не было технической возможности посмотреть на качество работы в случае модели огромной сложности
- С развитием техники стало возможным обучать модели с миллионами и даже миллиардами параметров
- Оказалось, что с увеличением сложности ошибка ведет себя сначала как предсказывает дилемма смещения-разброса, а затем неожиданно начинает снижаться⁶ и выходит даже на меньший уровень ошибки!
- Переломный момент — точка, в которой сложность модели сопоставима с мощностью обучающей выборки (interpolation threshold)
- Такое поведение называется **двойным спуском** (double descent)

⁶Advani, Madhu S., Andrew M. Saxe, and Haim Sompolsky. “High-dimensional dynamics of generalization error in neural networks.” 2017

Модель оптимальной сложности: двойной спуск

- Пример двойного спуска на практике⁷:



⁷Image source: <https://arxiv.org/pdf/1912.02292.pdf>

Спасибо за внимание!

