

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ВЫСШАЯ ШКОЛА ЭКОНОМИКИ

Факультет физики

ОАД

«Билеты к экзамену»

Лектор: Корнилов М. В.



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

Москва  
2021

# Содержание

<b>Замечания и благодарности</b>	<b>2</b>
<b>Структуры данных</b>	<b>3</b>
Билет 1-3. О-нотация. Приведите примеры алгоритмов над структурами данных с кон- стантным и квадратичным (с линейным и логарифмическим) по числу элементов временем работы. Приведите примеры, в которых выбор асимптотически более медленного с точки зрения О-нотации алгоритма предпочтителен. . . . .	3
Билет 4. Сортировка. Приведите примеры методов сортировки. . . . .	4
<b>Оптимизация</b>	<b>4</b>
Билет 1. Задачи оптимизации. Их классификация. Примеры . . . . .	4
<b>Машинное обучение</b>	<b>4</b>
Билет 1. Основная идея подхода к решению задач методами машинного обучения. Фундаментальное ограничение машинного обучения. . . . .	4
Билет 2-3. Классификация методов машинного обучения. Приведите примеры задач каждого класса. Задача классификации с учителем. Примеры. . . . .	5
Билет 4. Функция потерь в задачах машинного обучения. . . . .	7
Билет 5. Понятие переобучения. Тестовая выборка. . . . .	8
Билет 6. Идея метода k ближайших соседей (kNN). . . . .	8
Билет 7. Гиперпараметры и валидационная выборка. . . . .	9

## Замечания и благодарности

Данный конспект написан студентами и для студентов. Он может содержать опечатки, неточности и серьёзные смысловые ошибки. Над ним работали:

- Написание:

М. Блуменау

Ссылка на записи лекций

Ссылка на репозиторий с исходными файлами

## Структуры данных

**Билет 1-3. О-нотация. Приведите примеры алгоритмов над структурами данных с константным и квадратичным (с линейным и логарифмическим) по числу элементов временем работы. Приведите примеры, в которых выбор асимптотически более медленного с точки зрения О-нотации алгоритма предпочтителен.**

В информатике временная сложность алгоритма определяется как функция от длины строки, представляющей входные данные, равная времени работы алгоритма на данном входе. Временная сложность алгоритма обычно выражается с использованием нотации «О» большое, которая учитывает только слагаемое самого высокого порядка, а также не учитывает константные множители, то есть коэффициенты. Если сложность выражена таким способом, говорят об асимптотическом описании временной сложности, то есть при стремлении размера входа к бесконечности. Временная сложность обычно оценивается путём подсчёта числа элементарных операций, осуществляемых алгоритмом. Время исполнения одной такой операции при этом берётся константой, то есть асимптотически оценивается как  $O(1)$ . В таких обозначениях полное время исполнения и число элементарных операций, выполненных алгоритмом, отличаются максимум на постоянный множитель, который не учитывается в О-нотации.

Примеры алгоритмов:

- $O(1)$ , константный:

Определение чётности целого числа (представленного в двоичном виде)

- $O(n^2)$ , квадратичный:

Сортировка пузырьком (парное сравнение соседних элементов)

Сортировка вставками (рассматриваем по элементу по одному, каждый новый элемент - в подходящее место)

- $O(n)$ , линейный:

Поиск наименьшего или наибольшего элемента в неотсортированном массиве

- $O(\log(n))$ , логарифмический:

Бинарный поиск (метод деления пополам)

Пример, когда асимптотически более медленный алгоритм предпочтителен:

Сложность вычисления произведения матриц по определению составляет  $O(n^3)$ , однако существуют более эффективные алгоритмы, применяющиеся для больших матриц.

Первый алгоритм быстрого умножения больших матриц был разработан Фолькером Штрассеном в 1969. В основе алгоритма лежит рекурсивное разбиение матриц на блоки  $2 \times 2$ . Штрассен доказал, что такие матрицы можно некоммутативно перемножить с помощью семи умножений, поэтому на каждом этапе рекурсии выполняется семь умножений вместо восьми. В результате асимптотическая сложность этого алгоритма составляет  $O(n^{\log_2 7}) \approx O(n^{2.81})$ . Недостатком данного метода является большая сложность программирования по сравнению со стандартным алгоритмом, слабая численная устойчивость и большой объём используемой памяти. Разработан ряд алгоритмов на основе метода Штрассена, которые улучшают численную устойчивость, скорость по константе и другие его характеристики. Тем не менее, в силу простоты алгоритм Штрассена остаётся одним из практических алгоритмов умножения больших матриц.

В дальнейшем оценки скорости умножения больших матриц многократно улучшались. Однако эти алгоритмы носили теоретический, в основном приближённый характер. В силу неустойчивости алгоритмов приближённого умножения в настоящее время они не используются на практике. Лучший на данный момент -  $O(n^{2.37})$ .

Таким образом, можно сказать, что такие примеры в основном связаны с используемой памятью и устойчивостью.

## Билет 4. Сортировка. Приведите примеры методов сортировки.

### Оптимизация

## Билет 1. Задачи оптимизации. Их классификация. Примеры

### Машинное обучение

## Билет 1. Основная идея подхода к решению задач методами машинного обучения. Фундаментальное ограничение машинного обучения.

Машинное обучение – обширный подраздел искусственного интеллекта, изучающий методы построения алгоритмов, способных обучаться. Машинное обучение находится на стыке математической статистики, методов оптимизации и классических математических дисциплин, но имеет также и собственную специфику, связанную с проблемами вычислительной эффективности и переобучения. Многие методы разрабатывались как альтернатива классическим статистическим подходам.

Обучение по прецедентам, или индуктивное обучение, основано на выявлении общих закономерностей по частным эмпирическим данным. Дедуктивное обучение предполагает формализацию знаний экспертов и их перенос в компьютер в виде базы знаний. Дедуктивное обучение принято относить к области экспертных систем, поэтому термины машинное обучение и обучение по прецедентам можно считать синонимами.

Дано конечное множество прецедентов (объектов, ситуаций), по каждому из которых собраны (измерены) некоторые данные. Данные о прецеденте называют также его описанием. Совокупность всех имеющихся описаний прецедентов называется обучающей выборкой. Требуется по этим частным данным выявить общие зависимости, закономерности, взаимосвязи, присущие не только этой конкретной выборке, но вообще всем прецедентам, в том числе тем, которые ещё не наблюдались.

Наиболее распространённым способом описания прецедентов является признаковое описание. Фиксируется совокупность  $n$  показателей, измеряемых у всех прецедентов. Если все  $n$  показателей числовые, то признаковые описания представляют собой числовые векторы размерности  $n$ . Возможны и более сложные случаи, когда прецеденты описываются временными рядами или сигналами, изображениями, видеорядами, текстами, попарными отношениями сходства или интенсивности взаимодействия, и т. д.

Для решения задачи обучения по прецедентам в первую очередь фиксируется модель восстанавливаемой зависимости. Затем вводится функционал качества, значение которого показывает, насколько хорошо модель описывает наблюдаемые данные. Алгоритм обучения (learning algorithm) ищет такой набор параметров модели, при котором функционал качества на заданной обучающей выборке принимает оптимальное значение. Процесс настройки (fitting) модели по выборке данных в большинстве случаев сводится к применению численных методов оптимизации.

Выходом алгоритма обучения является функция, аппроксимирующая неизвестную (восстанавливаемую) зависимость. В задачах классификации аппроксимирующую функцию принято называть классификатором (classifier), концептом (concept) или гипотезой (hypothesis); в задачах восстановления регрессии — функцией регрессии; иногда просто функцией. В русскоязычной литературе аппроксимирующую функцию также называют алгоритмом, подчёркивая, что и она должна допускать эффективную компьютерную реализацию.

Практически всё обучение, которое мы ждём от компьютеров, состоит в сокращении информации до основных закономерностей, на основании которых можно делать выводы о чём-то неизвестном. Рассмотрим загадку:

$$1 + 4 = 5$$

$$2 + 5 = 12$$

$$3 + 6 = 21$$

$$8 + 11 = ?$$

Достаточно нетрудно заметить две закономерности:

$$1 * (4 + 1) = 5$$

$$2 * (5 + 1) = 12$$

$$3 * (6 + 1) = 21$$

И:

$$0 + 1 + 4 = 5$$

$$5 + 2 + 5 = 12$$

$$12 + 3 + 6 = 21$$

Какая же закономерность верна? Естественно, обе – и ни одна из них. Всё зависит от того, какие закономерности допустимы. Поиск закономерностей зависит от предположений наблюдателя.

То же верно и для МО. Даже когда машины обучают сами себя, предпочтительные закономерности выбираются людьми: должно ли ПО для распознавания лиц содержать явные правила если/то, или оно должно расценивать каждую особенность как дополнительное доказательство в пользу или против каждого возможного человека, которому принадлежит лицо? Какие особенности изображения должно обрабатывать ПО? Нужно ли ей работать с отдельными пикселями? Выбор подобных вариантов ограничивает то, какие закономерности система сочтёт вероятными или даже возможными. Эти вопросы ограничивают попытки применения нейросетей к новым задачам.

## **Билет 2-3. Классификация методов машинного обучения. Приведите примеры задач каждого класса. Задача классификации с учителем. Примеры.**

Обучение с учителем (supervised learning) — наиболее распространённый случай. Каждый прецедент представляет собой пару «объект, ответ». Требуется найти функциональную зависимость ответов от описаний объектов и построить алгоритм, принимающий на входе описание объекта и выдающий на выходе ответ. Функционал качества обычно определяется как средняя ошибка ответов, выданных алгоритмом, по всем объектам выборки.

- Задача классификации (classification) отличается тем, что множество допустимых ответов конечно. Их называют метками классов (class label). Класс — это множество всех объектов

с данным значением метки. Примеры: классификация цветов (датасет про ирисы), классификация частиц в последнем домашнем задании 2021 учебного года, классификация клиентов на платёжеспособных и нет.

- Задача регрессии (regression) отличается тем, что допустимым ответом является действительное число или числовой вектор. Примеры: стоимость квартиры, красное смещение в домашнем задании 5 2021 учебного года.
- Задача ранжирования (learning to rank) отличается тем, что ответы надо получить сразу на множестве объектов, после чего отсортировать их по значениям ответов. Может сводиться к задачам классификации или регрессии. Пример: ранжирование страниц в поиске в Интернете.
- Задача прогнозирования (forecasting) отличается тем, что объектами являются отрезки временных рядов, обрывающиеся в тот момент, когда требуется сделать прогноз на будущее. Пример: прогноз спроса на товар.

Обучение без учителя (unsupervised learning). В этом случае ответы не задаются, и требуется искать зависимости между объектами.

- Задача кластеризации (clustering) заключается в том, чтобы сгруппировать объекты в кластеры, используя данные о попарном сходстве объектов. Функционалы качества могут определяться по-разному, например, как отношение средних межкластерных и внутрикластерных расстояний. Пример: позитронно-эмиссионная томография для автоматического выделения различных типов тканей на трехмерном изображении.
- Задача поиска ассоциативных правил (association rules learning). Исходные данные представляются в виде признаковых описаний. Требуется найти такие наборы признаков, и такие значения этих признаков, которые особенно часто (неслучайно часто) встречаются в признаковых описаниях объектов. Пример: поиск ассоциативных правил в заданном наборе транзакций (хлеб, молоко и так далее).
- Задача фильтрации выбросов (outliers detection) — обнаружение в обучающей выборке небольшого числа нетипичных объектов. В некоторых приложениях их поиск является самоцелью. В других приложениях эти объекты являются следствием ошибок в данных или неточности модели, то есть шумом, мешающим настраивать модель, и должны быть удалены из выборки. Пример: обнаружение мошенничества.
- Задача построения доверительной области (quantile estimation) — области минимального объёма с достаточно гладкой границей, содержащей заданную долю выборки.
- Задача сокращения размерности (dimensionality reduction) заключается в том, чтобы по исходным признакам с помощью некоторых функций преобразования перейти к наименьшему числу новых признаков, не потеряв при этом никакой существенной информации об объектах выборки. В классе линейных преобразований наиболее известным примером является метод главных компонент.
- Задача заполнения пропущенных значений (missing values) — замена недостающих значений в матрице объекты–признаки их прогнозными значениями.

Это были главные, но ради интереса стоит указать:

Частичное обучение (semi-supervised learning) занимает промежуточное положение между обучением с учителем и без учителя. Каждый прецедент представляет собой пару «объект, ответ», но ответы известны только на части прецедентов. Пример: автоматическая рубрикация большого количества текстов при условии, что некоторые из них уже отнесены к каким-то рубрикам.

Обучение с подкреплением (reinforcement learning). Роль объектов играют пары «ситуация, принятое решение», ответами являются значения функционала качества, характеризующего правильность принятых решений (реакцию среды). Как и в задачах прогнозирования, здесь существенную роль играет фактор времени. Примеры: формирование инвестиционных стратегий, автоматическое управление технологическими процессами, самообучение роботов.

Динамическое обучение (online learning) может быть как обучением с учителем, так и без учителя. Специфика в том, что прецеденты поступают потоком. Требуется немедленно принимать решение по каждому прецеденту и одновременно доучивать модель зависимости с учётом новых прецедентов. Как и в задачах прогнозирования, здесь существенную роль играет фактор времени. Пример: системы автопилотов.

Активное обучение (active learning) отличается тем, что обучаемый имеет возможность самостоятельно назначать следующий прецедент, который станет известен.

Метаобучение (meta-learning или learning-to-learn) отличается тем, что прецедентами являются ранее решённые задачи обучения. Требуется определить, какие из используемых в них эвристик работают более эффективно. Конечная цель — обеспечить постоянное автоматическое совершенствование алгоритма обучения с течением времени.

Многозадачное обучение (multi-task learning). Набор взаимосвязанных или схожих задач обучения решается одновременно, с помощью различных алгоритмов обучения, имеющих схожее внутреннее представление. Информация о сходстве задач между собой позволяет более эффективно совершенствовать алгоритм обучения и повышать качество решения основной задачи.

Индуктивный перенос (inductive transfer). Опыт решения отдельных частных задач обучения по прецедентам переносится на решение последующих частных задач обучения. Для формализации и сохранения этого опыта применяются реляционные или иерархические структуры представления знаний.

Глубинное обучение может проходить как без учителя, так и с подкреплением. При глубинном обучении частично имитируются принципы обучения людей — используются нейронные сети для все более подробного уточнения характеристик набора данных. Глубинные нейронные сети применяются, в частности, для ускорения скрининга больших объемов данных при поиске лекарственных средств. Такие нейросети способны обрабатывать множество изображений за короткое время и извлечь больше признаков, которые модель в конечном счете запоминает. Глубинное обучение может использоваться в автомобильной отрасли при выполнении ремонта и профилактического обслуживания.

#### **Билет 4. Функция потерь в задачах машинного обучения.**

Предположим, дано задание наполнить мешок 5 кг муки. Вы заполняете его до тех пор, пока измерительный прибор не даст идеальное показание 5 кг, или вы достанете песок, если показание превысит 5 кг.

Точно так же, как весы, если ваши прогнозы не верны, ваша функция потерь будет выводить большее число. Если они довольно хороши, выведите меньшее число. Когда вы экспериментируете с вашим алгоритмом, чтобы попытаться улучшить свою модель, ваша функция потерь скажет вам, достигаете ли вы (или достигаете) где-либо.



Функция, которую мы хотим минимизировать или максимизировать, называется целевой функцией или критерием. Когда мы минимизируем его, мы можем также назвать его функцией стоимости, функцией потерь или функцией ошибки.

По своей сути функция потерь - это мера того, насколько хороша ваша модель прогнозирования с точки зрения возможности прогнозировать ожидаемый результат (или значение). Мы преобразуем задачу обучения в задачу оптимизации, определяем функцию потерь, а затем оптимизируем алгоритм, чтобы минимизировать функцию потерь.

Примеры:

- Mean Squared Error (MSE) - это рабочая область базовых функций потерь, так как она проста для понимания и реализации и в целом работает довольно хорошо. Чтобы рассчитать MSE, вы берете разницу между предсказаниями вашей модели и основополагающей правдой, вычеркиваете ее и затем усредняете по всему набору данных. Результат всегда положительный, независимо от знака предсказанных и основанных значений истинности, и идеальное значение равно 0,0.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

- Mean Absolute Error (MAE) - лишь немного отличается по определению от MSE, но, что интересно, обеспечивает почти совершенно противоположные свойства. Чтобы рассчитать MAE, вы берете разницу между предсказаниями вашей модели и основополагающей правдой, применяете абсолютное значение к этой разнице, а затем усредняете его по всему набору данных.

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n}$$

- Функция потерь Хьюбера — это функция потерь, используемая в устойчивой регрессии, которая менее чувствительна к выбросам, чем квадратичная ошибка.

$$\text{HuberLoss} = \begin{cases} \frac{1}{2}a^2 & \text{для } |a| \leq \delta, \\ \delta \left( |a| - \frac{1}{2}\delta \right) & \text{иначе.} \end{cases} \quad (1)$$

- Функция потерь для классификации - индикатор ошибки.

$$\text{ClassificationLoss} = \sum_{i=1}^n [a(x)_i \neq y(x)_i]$$

## Билет 5. Понятие переобучения. Тестовая выборка.

Тестовая (или контрольная) выборка (test sample) — выборка, по которой оценивается качество построенной модели. Если обучающая и тестовая выборки независимы, то оценка, сделанная по тестовой выборке, является несмещённой.

Оценку качества, сделанную по тестовой выборке, можно применить для выбора наилучшей модели. Однако тогда она снова окажется оптимистически смещённой. Для получения немещённой оценки выбранной модели приходится выделять третью выборку.

Переобучение, переподгонка (overtraining, 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 сложных моделей.

**Билет 6. Идея метода k ближайших соседей (kNN).**

Метод ближайших соседей (kNN - k Nearest Neighbours) - метод решения задач классификации и задач регрессии, основанный на поиске ближайших объектов с известными значениями целевой переменной. Метод основан на предположении о том, что близким объектам в признаковом пространстве соответствуют похожие метки. Для нового объекта метод предполагает найти ближайшие к нему объекты и построить прогноз по их меткам.

В случае использования метода для классификации объект присваивается тому классу, который является наиболее распространённым среди k соседей данного элемента, классы которых уже известны. В случае использования метода для регрессии, объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны.

Алгоритм может быть применен к выборкам с большим количеством атрибутов (многомерным). Для этого перед применением нужно определить функцию расстояния; классический вариант такой функции — евклидова метрика.

Разные атрибуты могут иметь разный диапазон представленных значений в выборке (например атрибут А представлен в диапазоне от 0.1 до 0.5, а атрибут Б представлен в диапазоне от 1000 до 5000), то значения дистанции могут сильно зависеть от атрибутов с большими диапазонами. Поэтому данные обычно подлежат нормализации. Некоторые значимые атрибуты могут быть важнее остальных, поэтому для каждого атрибута может быть задан в соответствии определённый вес (например вычисленный с помощью тестовой выборки и оптимизации ошибки отклонения). При взвешенном способе во внимание принимается не только количество попавших в область определённых классов, но и их удалённость от нового значения.

Гиперпараметры - это настраиваемые параметры, которые необходимо настроить, чтобы получить модель с оптимальными характеристиками. В случае kNN таковым является параметр k - числа соседей.

**Билет 7. Гиперпараметры и валидационная выборка.**

Гиперпараметры модели — параметры, значения которых задаются до начала обучения модели и не изменяются в процессе обучения. У модели может не быть гиперпараметров.

Параметры модели — параметры, которые изменяются и оптимизируются в процессе обучения модели и итоговые значения этих параметров являются результатом обучения модели.

Примерами гиперпараметров могут служить количество слоев нейронной сети, а также количество нейронов на каждом слое. Примерами параметров могут служить веса ребер нейронной сети. Для нахождения оптимальных гиперпараметров модели могут применяться различные алгоритмы настройки гиперпараметров.

Примеры: число деревьев в случайном лесе, параметр k для метода kNN, глубина деревьев в бэггинге.

Валидационная выборка (validation sample) — выборка, по которой осуществляется выбор наилучшей модели из множества моделей, построенных по обучающей выборке. Её объём может быть равен, например 0.1 от общего.

**Билет 8. Задача регрессии с учителем. Примеры. Популярные алгоритмы решения.**