

Титульный лист материалов по дисциплине
(заполняется по каждому виду учебного материала)

ДИСЦИПЛИНА	Технологии извлечения знаний из больших данных <small>(полное наименование дисциплины без сокращений)</small>
ИНСТИТУТ	ИКБ
КАФЕДРА	КБ-4 «Интеллектуальные системы информационной безопасности» <small>полное наименование кафедры)</small>
ВИД УЧЕБНОГО МАТЕРИАЛА	Лекция <small>(в соответствии с пп. I-III)</small>
ПРЕПОДАВАТЕЛЬ	Никонов В.В. <small>(фамилия, имя, отчество)</small>
СЕМЕСТР	3 семестр 2023/2024 уч. года <small>(указать семестр обучения, учебный год)</small>

Ассоциации, рекомендательные системы и обучение с подкреплением

Ассоциативные правила

Задачи классификации и регрессии концептуально похожи: в них нужно научиться предсказывать целевую переменную по признакам, только вид целевой переменной отличается, а значит и методы, решающие две задачи, немного разные, хотя и следуют общим идеям. Методы решения задач классификации и регрессии сегодня используются повсеместно в бизнесе и науке. Постановки задач обучения без учителя более разнообразны, а критерии качества не так очевидны, но и данные для этих задач нужны неразмеченные (без целевой переменной), а значит, раздобыть такие данные гораздо проще.

В задаче построения ассоциаций данные имеют вид набора транзакций, где каждая транзакция — это набор товаров из множества товаров. Например, транзакцией может быть набор продуктов, купленных покупателем при походе в магазин, а весь набор данных будет состоять из всех походов всех покупателей в этот магазин (или все магазины сети). Чаще всего методы поиска ассоциаций используется именно в торговле, но можно придумать примеры и в других областях. Важно, чтобы в каждой транзакции было не один и не два товара, а больше.

Задача поиска ассоциаций или ассоциативных правил, формулируется так: нужно найти закономерности вида «если покупатель купил <некоторый набор товаров>, то он купит и <еще один товар>» — эта закономерность и называется ассоциативным правилом.

Пример: «если покупатель купил зубную щетку, мыло и мыльницу, то он купит и чехол для зубной щетки». Как и в других задачах машинного обучения, суть задачи состоит в поиске правил по данным: конечно, сотрудники магазина могут знать о некоторых часто покупаемых вместе наборах товаров, но ни один сотрудник не сможет проанализировать столько чеков, сколько сможет компьютер.

Ассоциативные правила применяются для оптимизации размещения товаров на полках магазинов, для формирования персональных рекомендаций в онлайн-магазинах (хотя там больше используются рекомендательные системы, о которых мы поговорим ниже) и для планирования акций (можно сделать скидку на один товар, зная, что вместе с ним купят другой, более дорогой товар).

Поиск ассоциативных правил обычно выполняется в два этапа. На первом этапе находятся часто покупаемые наборы товаров: алгоритм проходит по всему набору данных много раз, вычисляет частоты различных комбинаций товаров и выделяет часто встречающиеся комбинации. На втором этапе происходит непосредственное выделение правил на основе найденных на первом этапе частых комбинаций. Самые известные алгоритмы поиска ассоциативных правил — Apriori и FP-Growth. Подробнее прочитать про поиск ассоциативных правил и указанные алгоритмы можно по [ссылке](#).

В больших наборах данных можно встретить самые разные комбинации товаров, и многие из них даже будут повторяться, поэтому очень важно найти правила, которые встретились действительно много раз, например сотни тысяч раз. Поэтому при поиске правил часто отслеживают следующие числовые показатели комбинаций товаров: **поддержка** (support) и **доверие** (confidence, переводят как достоверность или значимость).

- Поддержка — это частота, сколько раз комбинация товаров встретилась в чеках, разделить на общее число чеков. Например, в какой доле чеков встретилась комбинация «книга, журнал, ручка».

- Доверие определяется для правила, например «книга, журнал \leadsto ручка» и означает, в скольких процентах случаев, в которых купили книгу и журнал, также купили и ручку. На первом этапе поиска ассоциативных правил комбинации товаров фильтруются по поддержке, а на втором ищутся ассоциативные правила, превышающие порог на доверие.

Пороги частоты и доверия можно менять и получать разные результаты.

Например, рассмотрим набор из восьми транзакций и шести товаров для супермаркета.

ID	Мука	Разрыхлитель	Масло	Пиво	Хлеб	Молоко
1	1	1	1	0	0	0
2	0	0	1	0	1	1
3	0	0	0	1	1	0
4	1	1	1	0	0	1
5	1	1	0	0	0	0
6	0	0	1	1	0	0
7	1	1	1	0	1	1
8	0	1	0	1	1	0

По таблице можно заметить, что набор {Мука, Разрыхлитель, Масло} часто покупают вместе, можем подсчитать поддержку этого набора, разделив количество транзакций, содержащих этот набор (ID: 1, 4, 7), на общее количество транзакций:

$$\text{support}(\text{Мука}, \text{Разрыхлитель}, \text{Масло}) = \frac{3}{8}$$

Пиво и муку, например, по нашим данным, никогда не покупали вместе, поэтому поддержка этого набора будет равна 0.

Мы заметили, что набор {Мука, Разрыхлитель, Масло} обладает высокой поддержкой, попробуем вывести ассоциативное правило, например, {Мука, Разрыхлитель} \rightsquigarrow {Масло}. Вычислим доверие для этого набора, для этого нужно разделить поддержку {Мука, Разрыхлитель, Масло} на поддержку {Мука, Разрыхлитель}.

Как мы уже вычислили ранее,

$$\text{support}(\text{Мука}, \text{Разрыхлитель}, \text{Масло}) = \frac{3}{8} \quad \text{а}$$

$$\text{support}(\text{Мука}, \text{Разрыхлитель}) = \frac{4}{8}$$

(в транзакции 5 присутствует мука и разрыхлитель, но не масло)

Тогда
$$\text{confidence}(\text{Мука}, \text{Разрыхлитель} \leadsto \text{Масло}) = \frac{3/8}{4/8} = \frac{3}{4} = 0,75$$

. Таким образом, можно сделать вывод, что с большой вероятностью если покупатель купил муку и разрыхлитель, то он купит и масло. Возможно, покупатель собирается что-то испечь, но это к ассоциативным правилам уже не относится.

Чем выше доверие ассоциативного правила, тем более достоверно наше правило. Качество поиска ассоциативных правил обычно определяют, просматривая примеры найденных правил.

Программное обеспечение — язык Python и специальные библиотеки, например, Orange.

Рекомендательные системы

Ассоциативные правила позволяют обнаружить, какие товары покупают часто вместе с другими, на основе этого можно предложить клиенту, что ему еще купить. Такая постановка задачи гораздо более широкая и называется задачей построения рекомендательных систем, или задачей рекомендаций. По сравнению с ассоциативными правилами рекомендательные системы анализируют не только то, как часто товары покупают вместе, но и характеристики (признаки) товаров и клиентов, и позволяют сделать более качественные рекомендации. Хорошие рекомендации выгодны и продавцу, и клиенту: продавцу потому, что это увеличивает оборот, а клиенту — потому что он тратит меньше времени на поиск товаров и находит новые интересные товары. Кроме того, продавцу рекомендательные системы позволяют больше продавать редкие товары (без рекомендаций их бы никто не нашел).



Данные в задаче рекомендаций выглядят как таблица, по строкам которой клиенты, по столбцам — товары (фильмы, книги и т.д.), и для некоторых ячеек известна оценка клиента для товара (для большинства ячеек таблицы оценка не известна, поскольку каждый клиент, скорее всего, взаимодействовал только с небольшим количеством товаров). Значения в ячейках могут задавать явную оценку товара клиентом (например, если он нажал в интерфейсе на некоторое количество звездочек) или неявную, например, клиент зашел на страницу товара и не купил его, или пропустил в списке товаров и не зашел. Для клиентов и товаров могут быть известны некоторые признаки, как в других задачах машинного обучения. Задача ставится так: для каждого клиента мы хотим найти товары, которые он захочет купить, кроме тех, что он уже купил.

- **Контентный подход**

Самый простой способ построить рекомендательную систему — свести задачу к задаче регрессии. В данном случае объектом будет пара (клиент, товар), признаки — совокупность признаков товара и клиента, а целевая переменная — оценка товара клиентом. Для решения этой задачи регрессии можно использовать любые методы, например линейные или решающие деревья. Чтобы выполнить рекомендацию для клиента, мы выполним предсказания для всех пар (этот клиент, товар), то есть для всех товаров, и

выберем те, для которых предсказано наибольшее значение. Описанный подход называется контентным (Content-based).

- **Коллаборативная фильтрация**

Другой подход называется основанным на памяти (Memory-based), его еще называют коллаборативной фильтрацией. Его идею мы уже упомянули выше: чтобы порекомендовать клиенту товары, алгоритм выделяет товары, которые клиент уже купил, выделяет клиентов, купивших эти товары и находит товары, которые покупали эти клиенты. Такой подход активно использовался до 2009 года, после чего ему на смену пришли подходы, основанные на скрытых переменных.

- **Скрытые переменные**

Подходы, основанные на скрытых переменных, работают так: они создают новые признаковые описания для клиентов и для товаров и оценивают нужность товара клиенту на основе этих признаковых описаний. Про создание новых признаковых представлений мы подробно поговорили в предыдущем разделе: в рекомендациях выполняется примерно то же самое, но только с акцентом на поиск товаров для клиентов, и называются эти признаки скрытыми переменными. Именно системы этой группы сегодня чаще всего используются на практике.

Как правило, продвинутые рекомендательные системы выполняют предсказания не так быстро, как более простые, например контентные линейные методы. Поэтому на практике часто сначала применяют простую систему, чтобы выполнить первичный отсев товаров, а затем к оставшимся товарам применяют более сложную и качественную рекомендательную систему. Подробнее прочитать про рекомендательные системы и их использование на практике можно по [ссылке](#).

Качество в задачах рекомендаций измерить несложно, например, можно отложить часть данных в тестовую выборку и оценить, какое количество товаров, купленных пользователем, система смогла

спрогнозировать. Для создания рекомендательных систем часто используется система Spark.

Примеры использования рекомендательных систем

Одним из первых активно использовать рекомендательные системы стал американский интернет-магазин Amazon: еще в 2000-х годах там появился раздел «Покупатели, купившие этот товар, также купили...». Сам алгоритм, выбиравший список товаров, работал именно по тому принципу, который описан в заголовке: анализировал покупателей, купивших данный товар. За более чем 10 лет рекомендательные системы шагнули далеко вперед, и сейчас алгоритмы рекомендаций устроены гораздо сложнее и обеспечивают более трети всех продаж современных интернет-магазинов.

Рекомендательные системы используются повсеместно: в онлайн-магазинах, кинотеатрах, книжных магазинах, вы наверняка сами много раз с ними сталкивались. Например, магазин брендовой одежды и обуви BRANDSHOP с помощью товарных рекомендаций [увеличил](#) прибыль с сайта на 5,9%. В компании НОРБИТ использование специальных моделей по формированию персональных маркетинговых предложений для SMS-рассылок [позволило повысить](#) конверсию клиентов с 3% до 25%.

Задача обучения с подкреплением

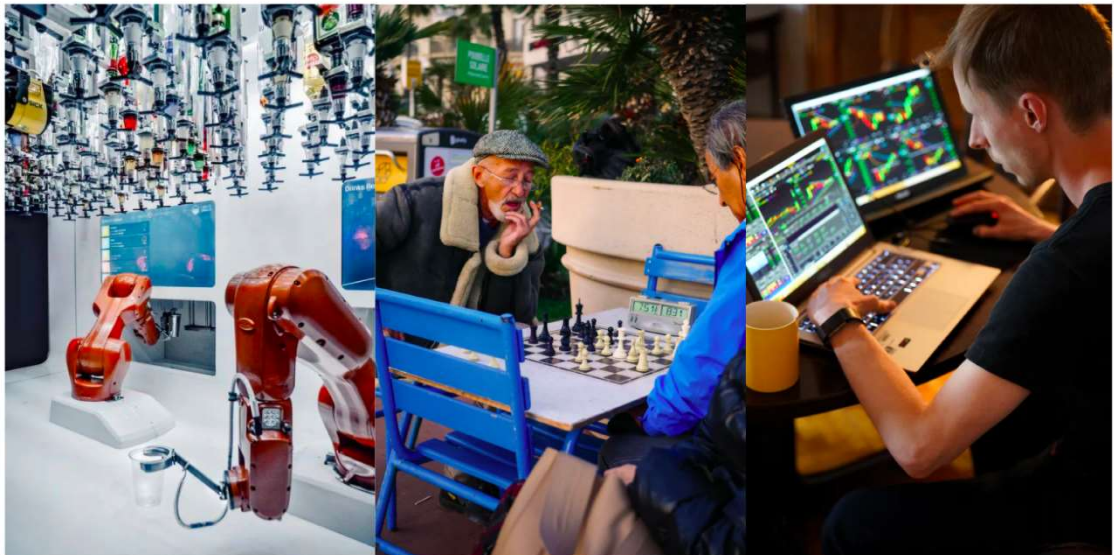
В задачах машинного обучения, в которых общая высокоуровневая схема работы выглядит так: **Сбор данных → Обучение алгоритма по данным → Применение готового алгоритма**. Это удобная и эффективная схема, повсеместно используемая при применении искусственного интеллекта к бизнес-задачам. Приведенная схема позволяет сначала внимательно обработать данные, затем аккуратно построить алгоритм на основе данных, и затем оценить качество работы алгоритма с использованием понятных метрик. Однако есть ряд задач, в которых такая схема с тремя отдельными шагами неэффективна, и сбор данных и обучение происходит прямо в процессе применения алгоритма. Область

искусственного интеллекта, изучающая такие задачи, называется обучением с подкреплением

Общая схема обучения с подкреплением

В обучении с подкреплением алгоритм обучается не на основе заранее собранных данных, а в процессе взаимодействия со средой. Сам алгоритм принято называть агентом. Примеры агентов и сред:

- Робототехника: агент — робот, среда — окружающий мир.
- Компьютерные игры: агент — игрок, среда — сама игра.
- Алгоритмический трейдинг: агент — трейдер (алгоритм), среда — биржа.



В обучении с подкреплением чередуются шаги действия агента и генерации награды средой. Например, если задача робота — забросить мяч в корзину, то его действия — это всевозможные движения «рук», а награды — очки за попадания в корзину. В алгоритмическом трейдинге действия — это операции с ценными бумагами, а награды — полученный доход или убыток. В компьютерных играх действия задаются правилами игры, награды — исходом игры (или промежуточными исходами игры). Награды в обучении с подкреплением могут быть отрицательными, то есть некоторые действия могут приводить к проигрышу, а не выигрышу.

Агент пробует выполнять разные действия, получает за них награды (выигрыши или проигрыши) и постепенно совершенствует свою стратегию,

то есть принципы, по которым агент выбирает действия. В обучении с подкреплением не только среда влияет на стратегию агента, но и действия агента могут менять среду, при этом говорят, что среда меняет состояние. Например, в компьютерных играх действия игрока (агента) могут менять конфигурацию игрового мира, эта конфигурация и будет задавать состояние среды.

Итак, схема обучения с подкреплением состоит из повторяющихся шагов:

- агент выполняет действие со средой;
- среда генерирует награду и (необязательно) обновляет состояние;
- агент корректирует стратегию.



Трейдинг как обучение с подкреплением

- Агент выбирает, какие бумаги и в каком количестве продать или купить;
- Награда равна потраченной или заработанной сумме; обновление среды соответствует изменению котировок и портфеля агента;
- На основе полученного нового опыта агент корректирует собственные правила, на основе которых покупает и продает бумаги.

В начале взаимодействия со средой агент не знает, какие действия выгоднее использовать, и выбирает, по сути, случайные действия. По мере

получения наград агент запоминает, какие действия и когда стоит использовать, и в дальнейшем получает больше наград. Даже если в какой-то момент в среде произойдут существенные изменения, благодаря постоянному совершенствованию стратегии агент сможет быстро адаптироваться и вновь получать хорошие награды.

Кумулятивная награда

Важной особенностью обучения с подкреплением является наличие отложенных наград: некоторые действия могут быть невыгодными в краткосрочной перспективе, но приносить большой выигрыш в долгосрочной перспективе. В примере с алгоритмическим трейдингом покупка каких-то акций может оказаться выгодным долгосрочным вложением, но упасть в цене в недалеком горизонте. В примере с роботом, бросающим мячи в корзину, замах руки назад направляет мяч в сторону, обратную от корзины, но помогает затем бросить мяч в нужную сторону с большей силой. В играх, например в шахматах, тоже иногда выгодно пожертвовать фигурой, чтобы впоследствии получить инициативу, более ценную фигуру или даже выиграть партию.

Наличие отложенных наград делает нетривиальным выбор стратегии. У агента, как правило, есть оценка того, к какой награде приведет каждое действие: в алгоритмическом трейдинге оценка очевидна, если известны текущие курсы акций; в других задачах она может быть менее очевидной. Однако жадная стратегия, то есть выбор самого прибыльного в текущий момент действия, может оказаться невыгодной в глобальной перспективе.

- В обучении с подкреплением обычно используют **кумулятивную награду**: разрабатывают стратегии, которые оптимизируют ожидаемую сумму всех последующих наград, а не только следующую.

- Часто также используют **дисконтирование**: при расчете кумулятивной награды далекую награду включают с меньшим весом, чем близкую. Дисконтирование будто бы комбинирует оба подхода: учитывает будущие награды, но уделяет больше внимания ближайшим наградам.

С понятием отложенного выигрыша также связано понятие **баланса исследования-использования** (exploration-exploitation trade-off).

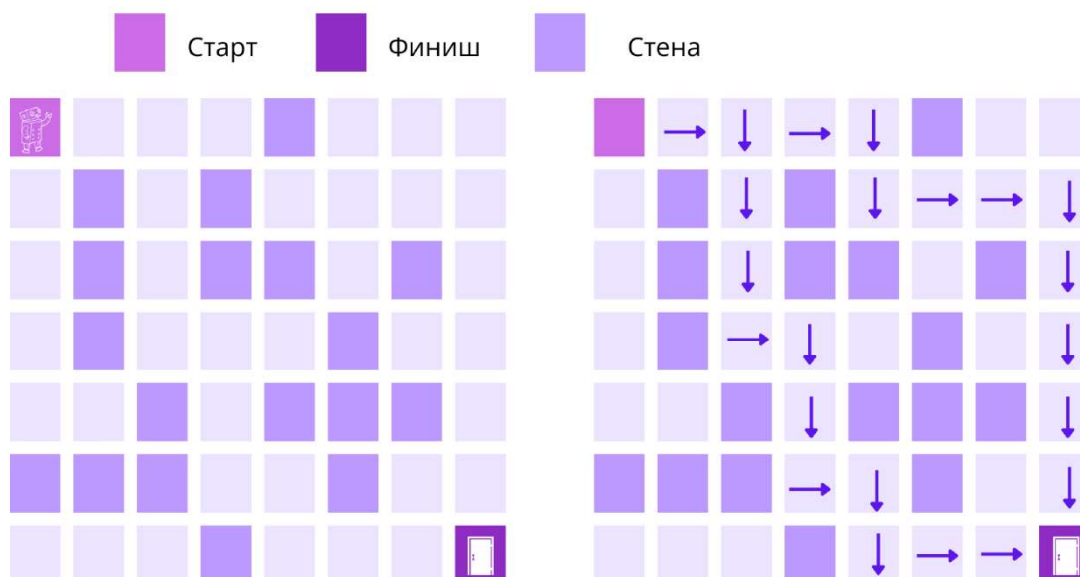
В начале работы агент пробует выполнять разные действия и запоминает, какие из них приносили награды в каких состояниях среды. После нескольких шагов агент попробовал пару-тройку действий, а про пользу или вред других действий агенту ничего не известно, потому что он их еще не пробовал. Далее агенту нужно выбрать следующее действие, и возникает выбор: либо выбирать наиболее оптимальное действие из ранее сделанных (оптимальность других действий оценить нельзя, так как агент их еще не пробовал выполнять), либо случайно выбрать из остальных, неизведанных действий.

Первый подход называется **стратегией использования** (exploitation), а второй — **стратегией исследования** (exploration). Понятно, что стратегия использования более надежная, а стратегия исследования — более рискованная: исследование потенциально может принести больший выигрыш, чем использование, но может оказаться и пустой тратой ресурсов. На практике между использованием и исследованием приходится балансировать.

Алгоритмы обучения с подкреплением

- Один из наиболее известных алгоритмов обучения с подкреплением называется Q-обучение (Q-learning). Немного упрощая, этот алгоритм учится предсказывать, какова будет кумулятивная награда от каждого действия в каждом возможном состоянии среды. Далее, зная текущее состояние среды, агент выбирает действие, которое принесет наибольшую кумулятивную награду. Q-обучение применимо в задачах, в которых количество возможных действий и состояний невелико.

Например, в простейшей игре «Лабиринт» у агента есть всего четыре действия (вперед, назад, влево, вправо), а состояние среды — это позиция агента в лабиринте.



- Если состояний среды может быть много, а действий все еще не очень много (например, в простых компьютерных играх, таких как Atari) то можно использовать нейронную сеть, которая по состоянию среды будет предсказывать кумулятивную награду от всех возможных действий — такой алгоритм называется Deep Q-Network (DQN). Среди концептуально похожих алгоритмов можно выделить алгоритм SARSA (State-Action-Reward-State-Action).

- Если количество возможных действий очень велико, используют подходы, в которых нейронная сеть учится предсказывать следующее действие по состоянию среды, например алгоритм REINFORCE. Наиболее успешными часто оказываются комбинированные подходы, в которых одна нейросеть предсказывает кумулятивную награду, другая — следующее действие, и при обучении они «помогают» друг другу. Пример такого комбинированного подхода — метод A3C (Asynchronous Advantage Actor-Critic). Например, A3C можно использовать в трейдинге — состояний биржи может быть бесконечно много (наборов котировок).

Применение обучения с подкреплением

Подведем итог: обучение с подкреплением — это отдельная область искусственного интеллекта, в которой агент учится выполнять выигрышные действия в результате взаимодействия со средой. Обучение с подкреплением — многообещающая технология, позволившая достичь высоких успехов ИИ в играх, однако в более прикладных сферах пока не достигшая уровня качества человека. Обучение с подкреплением является более сложной задачей, чем классическое обучение с учителем, поэтому если в бизнес-задачах можно использовать последнее, то лучше так и сделать.

Мы закончили темы, посвященные различным задачам машинного обучения. Как мы обсуждали ранее, понимание постановок задач машинного обучения важно для применения технологий искусственного интеллекта в своей области. Если заказчик сформулировал свою задачу как задачу классификации (указал, что будет объектом, какие признаки их описывают и какие классы нужно предсказывать), или как задачу рекомендаций (указал, какие есть товары и какая есть информация о предпочтениях клиентов, о самих товарах и клиента), или как задачу обучения с подкреплением (указал агента, среду, действия и награды), то специалисту по машинному обучению будет сразу понятно, какие методы применять для решения задачи. Конечно, бывают и нестандартные задачи, их решение обычно требует больше времени и затрат.