Теоретический минимум по машинному обучению (Поток Воронцов К.В. и Грабовой А.В.)

Константин Вихлянцев
Андрей Цедрик
Алексей Алексеев
Иван Мартынов

28 мая 2025 г.

1 Решающие деревья

1. **Что такое решающее дерево и как оно работает?** Решающее дерево – это модель, которая рекурсивно разбивает множество объектов на подмножества, принимая решения в узлах на основе признаков, чтобы предсказать класс (или значение) в листьях.

2. **Какие критерии для выбора параметров в узлах дерева, какой из какого следует и почему?** Для классификации применяются энтропия и индекс Джини (оба измеряют неоднородность классов), а для регрессии – среднеквадратичная ошибка (MSE); выбор критерия зависит от задачи, но все они направлены на уменьшение неоднородности в поддеревьях.

3. **Какие гиперпараметры регулируют сложность решающего дерева?** Глубина дерева (max_depth), минимальное количество объектов в узле для разделения (min_samples_split), минимальное количество объектов в листе (min_samples_leaf) и максимальное количество признаков при разбиении (max_features).

4. **Как решающие деревья обрабатывают задачи регрессии?** Дерево предсказывает среднее значение целевой переменной среди объектов в листе и строится с целью минимизации MSE.

5. **В чем разница между классификационным и регрессионным деревом?** Классификационное дерево предсказывает категории и использует меры неоднородности (энтропию, индекс Джини), а регрессионное – числовые значения и минимизирует среднеквадратичную ошибку (MSE).

2 Логистическая регрессия

1. **Что такое логистическая регрессия и какую задачу она решает?** Логистическая регрессия – это линейная модель для бинарной классификации, которая предсказывает вероятность принадлежности к классу через сигмоиду:

$$\sigma(z) = \frac{1}{1 + e^{-z}},$$

где $z = \beta_1 \cdot x_1 + \beta_2 \cdot x_2 + ... + \beta_n \cdot x_n + b$.

2. **Как выглядит функция потерь в логистической регрессии?** Используется логарифмическая функция потерь (логлосс):
$L = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log \hat{p}_i + (1 - y_i) \log(1 - \hat{p}_i)]$,
где $\hat{p}_i$ – предсказанная вероятность для объекта i.

3. **Как оптимизируются параметры в логистической регрессии?** Через градиентный спуск или его варианты (Adam, SGD), так как логлосс – выпуклая функция, но аналитического решения, как в линейной регрессии, нет.

4. **Как обобщить логистическую регрессию на многоклассовую классификацию?** OvR (One-vs-Rest): обучается одна модель на каждый класс против остальных. Softmax-регрессия (многоклассовая логистическая): одна модель сразу предсказывает вероятности для всех классов через softmax-функцию (обобщение сигмоиды).
Формула softmax для K классов:

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}},$$

где:

- z_i – логит (выход модели до применения softmax) для класса i;
- $\sum_{j=1}^{K} e^{z_j}$ – сумма экспонент всех логитов.

5. **Чем логистическая регрессия отличается от линейной?** Линейная регрессия предсказывает непрерывные значения, логистическая – вероятности классов, используя сигмоиду и логлосс вместо MSE.

6. **Какие метрики используются для оценки классификации?**

   - Accuracy: доля верно предсказанных объектов.
   - Precision: из всех предсказанных положительных – сколько верных?
   - Recall: из всех настоящих положительных – сколько найдено?
   - F1-score: гармоническое среднее между precision и recall.
   - Logloss: средняя логарифмическая ошибка – штрафует за неправильные уверенные предсказания.

7. **Что такое матрица ошибок?** Это таблица 2×2 (или n×n) где по строкам – истинные классы, по столбцам – предсказанные, показывающая, как модель путает классы.

3 SVM (Support Vector Machines)

1. **В чем основная идея SVM?** Основная идея SVM – найти такую гиперплоскость, которая максимально разделяет классы, то есть имеет наибольший зазор (margin) между ближайшими точками двух классов.

2. **Как строится разделяющая гиперплоскость в SVM?** SVM находит гиперплоскость, которая максимизирует отступ до ближайших объектов каждого класса (опорных векторов), решая задачу квадратичной оптимизации.

3. **Зачем в SVM используются ядра?** Когда классы не разделимы линейной границей, мы можем отобразить данные в более высокоразмерное пространство, где они станут линейно разделимыми. Это делается неявно с помощью ядровой функции $K(x, x')$, которая вычисляет скалярное произведение в этом новом пространстве, не создавая его явно (kernel trick).

4. **Назовите основные ядра в SVM.**
   - Линейное (linear);
   - Полиномиальное (polynomial);
   - Радиально-базисное (RBF, Gaussian);
   - Сигмоидное (sigmoid).

5. **Как SVM обобщается для задач регрессии?** Support Vector Regression (SVR). Вместо того чтобы находить гиперплоскость, разделяющую классы, SVR ищет функцию, максимально приближенную к данным, но допускает небольшие отклонения – в пределах заданного параметра ε (эпсилон). То есть модель не наказывается за небольшие ошибки, а штрафует только те, что выходят за ε-интервал.

6. **Как работает метод опорных векторов для несбалансированных данных?** Для несбалансированных данных можно использовать взвешивание классов: увеличить штраф за ошибку на меньшинстве, чтобы модель не игнорировала малочисленный класс при обучении.

4 Линейная регрессия

1. **Как работает линейная регрессия?** Линейная регрессия моделирует зависимость целевой переменной от признаков как линейную комбинацию признаков: $\hat{y} = X\beta + b$, где параметры подбираются для минимизации ошибки предсказания.

2. **Как находится оптимальное значение параметров?** Оптимальные параметры находятся методом минимизации среднеквадратичной ошибки (MSE), обычно через аналитическое решение (МНК) или численно, например градиентным спуском.

3. **В чем проблема мультиколлинеарности и как ее решают**? Мультиколлинеарность возникает, когда признаки линейно зависимы, из-за чего коэффициенты становятся нестабильными; решается с помощью регуляризации (Lasso, Ridge) или удаления/объединения коррелирующих признаков.

4. **Объясните вероятностную интерпретацию регуляризации в случае Lasso и Ridge.** Lasso соответствует лапласовскому априорному распределению на коэффициенты (более "острый"пик в нуле, стимулирующий обнуление коэффициентов), а Ridge – гауссовскому априорному распределению (нулевое среднее, нормальное распределение).

5. **Чем Lasso отличается от Ridge регуляризации?** Lasso использует L_1-норму и может занулять коэффициенты, выполняя отбор признаков; Ridge использует L_2-норму и только уменьшает веса, не обнуляя их.

6. **Как оценивать качество регрессионных моделей?** Основные метрики: MSE, RMSE (Root MSE), MAE (средняя абсолютная ошибка).

5 Ансамблирование моделей

1. **Назовите основные виды ансамблирования и их основное отличие.**
   - Бэггинг – строит модели независимо и усредняет (уменьшает дисперсию);
   - Бустинг – строит модели последовательно, каждая исправляет ошибки предыдущей (уменьшает смещение);
   - Стекинг – обучает метамодель на предсказаниях базовых моделей.

2. **Когда лучше использовать бэггинг, а когда — бустинг?**
   - Бэггинг – когда модель переобучается (например, решающие деревья);
   - Бустинг – когда модель недообучена и нужно снизить смещение.

3. **Приведите примеры алгоритмов бустинга.**
   - Gradient Boosting
   - XGBoost

4. **Приведите примеры алгоритмов бэггинга.**
   - Random Forest
   - Extra Trees

5. **Как работает алгоритм Random Forest?** Random Forest – это ансамбль решающих деревьев, обученных на разных подвыборках данных и случайных подмножествах признаков; итоговое предсказание – голосование (классификация) или усреднение (регрессия).

6. **Объясните принцип работы XGBoost.** XGBoost строит деревья последовательно, минимизируя дифференцируемую функцию потерь с регуляризацией; использует продвинутые оптимизации (обрезку деревьев, параллелизм, кэширование) для высокой производительности.

7. **Что такое "ансамблевое усреднение"?** Это способ объединения предсказаний нескольких моделей путем голосования (для классификации) или простого усреднения (для регрессии).

8. **Как работает метод Stacking в ансамблировании?** Stacking обучает несколько моделей (базовый уровень), собирает их предсказания, а затем обучает финальную модель (мета-уровень) на этих предсказаниях.

9. **Как работает метод случайного подпространства (Random Subspace) в бэггинге?** Каждая модель обучается не только на случайной подвыборке объектов, но и на случайном подмножестве признаков, что повышает разнообразие и устойчивость ансамбля.

6 Отбор признаков

1. **Какие методы отбора признаков вы знаете?**

   - Фильтрационные методы работают независимо от модели и используют статистические меры для оценки значимости признаков.
   - Обёрточные методы оценивают подмножества признаков на основе качества модели, что обеспечивает высокую точность, но увеличивает вычислительную сложность (например, полный перебор).
   - Эвристические и случайные методы используют эвристики или случайный подход для нахождения подмножеств признаков, что снижает вычислительную сложность (например, генетический алгоритм).

2. **Почему L1-регуляризация зануляет некоторые параметры?** В L1-регуляризации допустимая область решений – это ромб (в 2D). А уровни функции потерь (например, квадратичной) – это эллипсы. Оптимум находится в точке касания эллипса и ромба. Чаще всего это угловая точка ромба, где один из коэффициентов равен нулю.

3. **Как работает жадный алгоритм Add-Del для отбора признаков?**

   (a) Хранит и пытается заполнить множество оптимальных признаков. Состоит из двух фаз:

i. add: в мн-во добавляется признак, максимально улучшающий метрику эффективности, улучшение должно быть выше порога (по очереди перебираются все признаки и переобучается модель);

ii. del: то же, но из оптимального набора удаляется признак, наименее сильно ухудшающий модель.

(b) Переключение фаз либо по количеству шагов на каждой из них, либо по порогу изменения метрики.

(c) Остановка если модель не меняется при применении фаз (уже все найдено), если набрано нужное кол-во признаков, если пройдено макс кол-во итераций, если начинается переобучение

4. **Объясните критерий Фишера для отбора признаков.** Метод отбора признаков, используемый в задачах классификации. Его основная идея заключается в оценке, насколько хорошо каждый отдельный признак способен разделять данные на различные классы. Для 2 классов: $F = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$, где используются средние значения и стд признака по соответствующим классам. Если разница среднего большая => признак полезный. Для нескольких классов $F = \frac{\Sigma_j n_j (\mu_j - \mu)^2}{\Sigma_j n_j \sigma_j^2}$, n - кол-во элементов в классе, μ - по всей выборке.

5. **Как метод Белсли и SVD помогают бороться с мультиколлинеарностью?**

(a) Синглуярное разложение: если одно или несколько сингулярных значений близки к нулю, это указывает на то, что столбцы матрицы X (т.е. признаки) почти линейно зависимы. Чем меньше сингулярное значение, тем сильнее коллинеарность. Вместо регрессии на исходные коррелированные признаки, регрессия строится на наборе некоррелированных (ортогональных) главных компонент. Главные компоненты – это линейные комбинации исходных признаков, которые соответствуют правым сингулярным векторам

(b) Метод Белсли считает индексы обусловленности (отношение максимального сингулярного значения к каждому сингулярному значению):

i. если индексы большие (обычно > 30), то есть серьезные проблемы с мультиколлинеарностью; если маленькие их скорее всего нет, если очень большие (>100) то мультиколлинеарность почти идеальная.

ii. Для интерпретации сингулярных чисел используют доли разложения дисперсии: показывают, какая часть дисперсии оценки каждого коэффициента регрессии связана с каждым сингулярным значением (и, следовательно, с каждым индексом

обусловленности). Если высокий индекс обусловленности ассоциируется с высокими долями разложения дисперсии (например, > 0.5) для двух или более коэффициентов регрессии, это является сильным свидетельством того, что эти переменные вовлечены в коллинеарную связь, породившую данный высокий индекс.

7 Деление выборки

1. **В чем проблемы случайного деления выборки на обучение и контроль?**

   (a) Нет гарантии, что обучающая и тестовая выборки будут в полной мере представлять общую совокупность данных;

   (b) При случайном разбиении в тестовую или обучающую выборку может попасть недостаточное количество примеров редкого класса;

   (c) Утечка данных: при работе с данными, имеющими временную или групповую структуру, могут утечь данные из теста в обучение (например, во временных данных случайное разбиение может поместить будущие данные в обучающую выборку, что даст модели информацию из "будущего", а на "реально будущем" такой информации нет и модель плохо себя покажет).

2. **Как стратификация решает проблему несбалансированных данных?** Это метод разделения набора данных, который гарантирует, что пропорциональное представительство определенных характеристик (страт) сохраняется в каждой создаваемой подвыборке.

   (a) Весь набор данных делится на непересекающиеся подгруппы, называемые стратами. Они формируются на основе характеристики, которая считается важной для сохранения пропорций в конечных выборках (например принадлежность к классу).

   (b) Из каждой страты для обучающей и тестовой выборок производится независимая случайная выборка этого количества примеров.

3. **Что такое утечка данных при разделении выборки?** См 2 вопроса выше, примеры (кроме времени): есть данные о покупках разных клиентах, если поделить клиента между тестом и трейном, в модель будет зашита информация о его покупках что повысит результат теста, а на реальных данных не сработает; использование мат ожидания и стд, полученных по полной выборке, тоже дает информацию из теста.

8 Нейронные сети

1. **Что такое полносвязная нейронная сеть?** Сеть состоит из слоев нейронов, где каждый нейрон одного слоя соединен со всеми нейронами предыдущего слоя и передает свой выходной сигнал всем нейронам следующего слоя. Нейрон принимает взвешенную сумму входных сигналов и применяет к ней функцию активации.

2. **Назовите популярные функции активации, в чем их отличие.**

   (a) Сигмоида $\frac{1}{1+e^{-x}}$ и tanh $\frac{e^{x}+e^{-x}}{e^{x}+e^{-x}}$: сжимают выход в (-1, 1), страдают от проблемы исчезающего градиента;

   (b) RELU $max(0, x)$, leaky RELY $max(ax, x), a << 1$: полулинейные или кусочно-линейные, решают проблему исчезающего градиента для положительных значений, вычислительно эффективны, leaky решает проблемы исчезающего градиента;

   (c) softmax: применяется не к отдельному нейрону, а к целому слою (обычно выходному) и используется в основном на выходном слое для задач многоклассовой классификации.

3. **Какие проблемы возникают в полносвязных сетях?**

   (a) Большое количество параметров (долго обучать и возможно переобучение);

   (b) Обрабатывают входные данные как плоский вектор, полностью игнорируя их исходную структуру;

   (c) Проблема "исчезающего/взрывающегося градиента" (обучение нестабильное/медленное);

   (d) Черный ящик (непонятно как работает).

4. **Что такое dropout, чем отличается в обучении и в контроле?**

   (a) в обучении: для каждого слоя, к которому применяется dropout, для определенного процента нейронов их активации временно устанавливаются равными нулю;

   (b) в контроле: никто не выключается, но масштабируются коэффициенты на $(1 - dropout\_rate)$.

5. **Что такое BatchNorm, чем отличается в обучении и в контроле?**

   (a) BatchNorm нормализует входные данные каждого слоя, преобразуя их таким образом, чтобы они имели среднее ≈ 0 и стд ≈ 1. В отличие от нормализации ***входных*** данных, BatchNorm делает это динамически для выходов каждого слоя во время обучения;

(b) В обучении: считаем среднее и стд для каждого нейрона в текущем мини-батче (запоминаем также их скользящее среднее), нормализуем и сдвигаем с помощью двух обучаемых параметров: гамма и бета: $y = \gamma x + \beta$;

(c) В контроле: то же, но используем посчитанное скользящее среднее и обученные коэффициенты.

6. **Что такое паралич нейронной сети и как его избежать?**

   (a) Градиент функций активаций маленький (для sigma и tanh актуально), из-за чего коэффициенты почти не обновляются.

   (b) Можно заменить активацию на ReLU и подобное.

   (c) Нормализация входных данных или по батчу может сдвинуть значения ближе к нулю.

   (d) Использование оптимизаторов с адаптивной скоростью обучения, таких как Adam и RMSprop (подстраивают скорость обучения для каждого параметра индивидуально).

   (e) Правильная инициализация весов.

   (f) Уменьшение размера шага обучения (замедлит, но может убрать паралич).

7. **Опишите идею обратного распространения ошибки в нейросетях.**

   (a) Прямой проход: обычная работа сети;

   (b) Подсчет лосс функции;

   (c) Обратный проход: ошибка, вычисленная на выходном слое, "распространяется" обратно через сеть. На каждом слое вычисляется "вклад" каждого нейрона в эту ошибку. Для вычисления того, как изменение каждого отдельного веса влияет на конечную ошибку, используется цепное правило их исчисления (для подсчета градиента).

8. **Опишите идею прямого метода дифференцирования в нейросетях.** Для каждой входной переменной, по которой мы хотим найти производную, мы приписываем ей "тангенциальное" значение, которое показывает направление дифференцирования (как правило, 1 для переменной, по которой дифференцируем, и 0 для остальных). Затем, по мере того как данные проходят через вычислительный граф нейронной сети (от входа к выходу), мы одновременно вычисляем как значение функции в каждом узле, так и значение производной в этом узле.

9. **Что такое градиентный спуск и его варианты?** Градиент указывает направление наискорейшего роста функции. Градиентный спуск

использует это свойство: алгоритм итеративно движется в направлении, противоположном градиенту в текущей точке (в направлении наискорейшего убывания функции). Каждый шаг приближает к минимуму функции.
Варианты:

(a) Пакетный градиентный спуск: на каждом шаге градиент вычисляется по всему обучающему набору данных;

(b) Стохастический градиентный спуск: на каждом шаге градиент вычисляется по одному случайно выбранному обучающему примеру;

(c) Мини-пакетный градиентный спуск (очевидно);

(d) Градиентный спуск с моментом: добавляет к текущему градиенту скользящее среднее прошлых значений.

9 RNN, LSTM, GRU

1. **Как работают рекуррентные нейронные сети?** Скрытое состояние нейрона (или слоя) на текущем шаге времени передается как дополнительный вход на следующем шаге времени: сеть сохраняет внутреннее состояние, которое служит своего рода краткосрочной памятью о предыдущих элементах последовательности.

2. **Какие проблемы возникают при обучении RNN?** Трудности с обучением долгосрочных зависимостей. Например, из-за маленького градиента, веса, с более ранних шагов времени, обновляются незначительно и сеть не учится на основе давней информации; также слишком большой градиент и нестабильность ниже описаны в вопросе про gradient clipping.

3. **Как LSTM и GRU решают эти проблемы и какие проблемы?**

   (a) LSTM решают проблемы RNN благодаря введению состояния ячейки (cell state) (в добавок к скрытому состоянию (hidden)) и трех типов вентилей:

      i. входного (решает, какую новую информацию следует "запомнить");

      ii. забывающего (решает, какую информацию из предыдущего состояния ячейки следует "забыть");

      iii. выходного (решает, какую часть текущего состояния ячейки следует использовать в качестве скрытого состояния на следующем шаге).

   (b) GRU: упрощенный LSTM без cell state (только hidden) и только два вентиля

i. Вентиль обновления – решает, какая часть предыдущего скрытого состояния должна быть сохранена, а какая часть нового кандидатного скрытого состояния должна быть добавлена;

ii. Вентиль сброса – игнорирует часть предыдущего состояния при подсчете нового.

4. **Что такое Gradient Clipping и зачем он нужен?** Для борьбы с проблемой взрывающихся градиентов (градиенты могут резко возрастать в процессе обратного распространения ошибки во времени или через многочисленные слои, вызывая проскакивание минимума) – просто ограничим значение координат градиента $[-C, C]$, либо нормируем сразу весь вектор чтобы его длина была меньше указанной.

10 Сверточные сети (CNN)

1. **В чем идея сверточных нейронных сетей?** Вместо обработки изображения попиксельно, CNN применяют к нему набор локальных фильтров (сверток) для всех его точек, что использует локальную связность (пиксели имеют отношение только к своим соседям) и инвариантность к сдвигу (одна и та же матрица везде применяется).

2. **Что такое операция свертки, какие у нее есть параметры?** Для всех позиций на изображении высчитывается взвешенное среднее пикселей вокруг, где веса берутся из обучаемой матрицы.
Параметры:

   (a) Kernel Size – размер матрицы и кол-во пикселей которые берутся вокруг;

   (b) Шаг (stride) – количество пикселей, на которое сдвигается фильтр после каждого применения;

   (c) Отступ (padding) – определяет, добавляются ли нулевые пиксели по границам входных данных перед выполнением свертки;

   (d) Входное/выходное кол-во каналов (для каждого канала своя матрица);

   (e) dilation – расстояние между пикселями в свертке.

3. **Какие слои кроме сверточных используются в** CNN**?**

   (a) Макс-пулинг (Max Pooling): из каждого “окна” (небольшой области) на входной карте признаков выбирается максимальное значение;

   (b) Средний пулинг (Average Pooling): из каждого “окна” вычисляется среднее значение;

   (c) Полносвязные слои после конвертации в 1D вектор.

4. **В чем особенность архитектуры ResNet?** В использовании так называемых остаточных соединений (residual connections) или пропускных соединений (skip connections). Это сделано из-за проблемы деградации (отсутствие улучшения или даже ухудшение модели) при увеличении количества слоев из-за трудностей оптимизации очень глубоких структур, в частности, с проблемой затухания градиентов при обратном распространении ошибки. Идея остаточных соединений: вместо того чтобы каждый слой пытался напрямую выучить требуемое преобразование из входа в выход (H(x)), ResNet предлагает, чтобы слои выучивали остаточное отображение ($F(x) = H(x) - x$). Иными словами, слои учатся предсказывать разницу между желаемым выходом и входом блока. Выход такого "остаточного блока" тогда будет равен $F(x) + x$. Это решает: легче обучать преобразования близкие к тождественным. Пропускные соединения создают "прямые пути" для распространения градиентов во время обратного распространения ошибки.

5. **Как дообучать предобученные модели под конкретную задачу?** Т. к. модель основные паттерны запомнила, достаточно заменить только последний(-ие) слои. Дообучать можно только свои крайние слои (данных мало и цели похожи), часть или все слои (в противном случае).

11 Автокодировщики и GAN

1. **Что такое автокодировщик и зачем он нужен?** Это нейросеть, которая обучается без учителя и используется для повышения или понижения размерности данных с их последующим восстановлением, выделения наиболее важных признаков. Состоит из энкодера, который переводит данные в латентное представление (latent space), и декодера, который восстанавливает данные из латентного представления.

2. **Как линейный автокодировщик связан с PCA?** PCA (метод главных компонент) – это статистический метод снижения размерности данных, который преобразует признаки в новый набор ортогональных (независимых) переменных, называемых главными компонентами. Линейный автокодировщик с MSE-функцией потерь и линейными активациями математически эквивалентен PCA, так как оба метода находят оптимальное линейное подпространство для минимизации ошибки реконструкции. Разница лишь в способе оптимизации: PCA использует аналитическое решение (SVD (сингулярное разложение матрицы)), а автокодировщик – градиентный спуск.
tl;dr Линейный автокодировщик – это нейросетевой аналог PCA, но с возможностью обобщения на нелинейные случаи.

3. **Опишите принцип работы вариационного автокодировщика.** Дальше будет описание одной из наших задач. VAE состоит из энкодера и декодера: на входе x (input dim), внутри z и его параметры нормального распределения μ и σ (latent dim), на выходе параметры распределения x ($2 * inputdim$ в случае нормального распределения, т.к. параметров два). Получаем на вход x, с помощью линейной модели преобразуем в параметры распределения z, потом сэмплируем z (то есть получаем z из его параметров распределения), отсюда с помощью линейной модели получаем предсказанные параметры распределения x. Можно генерировать объекты, похожие на изначальные x, сэмплируя их из предсказанных параметров распределения. Функция потерь (ELBO) преследует цель максимизации нижней оценки log-правдоподобия: $\mathcal{L}_{VAE} = \mathcal{L}_{reconstruction} + \mathcal{L}_{KL}$ (ошибка реконструкции + KL-дивергенция).

4. **Чем GAN отличается от автокодировщика?** GAN (генеративно-состязательная сеть) и автокодировщик решают разные задачи. GAN генерирует новые данные через соревнование генератора и дискриминатора: генератор создает fake-данные из шума (например, изображения, фото людей), а дискриминатор пытается отличить fake от реальных данных (цель – соревнование, где генератор учится обманывать дискриминатор). Автокодировщик сжимает и восстанавливает данные, полезен для очистки или снижения размерности. Генерация в GAN качественнее, но обучение сложнее, тогда как автокодировщики стабильнее, но выдают размытые результаты.

12 Обучение с подкреплением (RL)

1. **В чем основная идея обучения с подкреплением?** Это метод, где агент учится принимать решения через взаимодействие со средой, получая награды за правильные действия. Цель – максимизировать совокупную награду, балансируя между исследованием новых действий и эксплуатацией известных стратегий. Можно сказать, что это частный случай обучения с учителем, где среда является неявным учителем, воздействующим на агента через слабую обратную связь (награды). Примеры: игры, робототехника, управление автономными системами.

2. **Приведите пример задачи RL с байесовскими бандитами.** Задача байесовских бандитов – это упрощенная модель RL, где агент выбирает между несколькими “однорукими бандитами” (слотами), каждый с неизвестным распределением наград. Используя байесовский подход, агент обновляет свои представления о вероятностях выигрыша и балансирует между исследованием (exploration) и эксплуатацией (exploitation).
Постановка задачи: есть K бандитов, каждый выдает награду r из

своего распределения (например, Бернулли: $r\ Bernoulli(p_i)$, где p_i – неизвестная вероятность успеха). Цель агента: максимизировать суммарную награду за T шагов.
Пример: динамический выбор рекламного баннера с максимальной конверсией. Бандиты – это две версии сайта (A и B), а награда – конверсия (1 – клик, 0 – нет).

3. **Как решается задача заплыва?** Задача решается с помощью алгоритмов обучения с подкреплением, где агент управляет телом с суставами в вязкой среде и обучается генерировать движения, обеспечивающие продвижение вперёд. Состояние включает углы и скорости суставов, действия – крутящие моменты. Агент получает награду за скорость или расстояние, пройденное вперёд, и со временем оптимизирует свою стратегию действий. Для обучения используются алгоритмы:

    - Policy Gradient – обучает стратегию напрямую, улучшая вероятность полезных действий на основе полученных наград.
    - PPO (Proximal Policy Optimization) – улучшенная версия policy gradient, которая ограничивает слишком резкие обновления стратегии.
    - Actor-Critic – сочетает две модели: actor (предлагает действия) и critic (оценивает их качество).

13 Активное обучение

1. **Какие проблемы решает активное обучение?** Активное обучение решает проблемы нехватки размеченных данных, высокой стоимости разметки, небалансированных классов (если, например, какие-то классы в данных встречаются очень редко), неопределенности модели (модель не уверена).
Пример с обучением алгоритма для робомобиля: специальная нейросеть обучается на уже размеченных данных, после этого обрабатывает неразмеченные данные, выбирая кадры, которые она не может распознать – таким образом, она ищет данные, которые будут представлять трудность для целевого алгоритма. Затем эти данные изучаются и размечаются людьми, и добавляются в базу обучающих данных.

2. **Назовите методы активного обучения.**

    - Uncertainty Sampling (выбор по неопределенности) – запрашивает примеры, в которых модель менее всего уверена.
    - Query-by-Committee (QBC, запрос по комитету) – несколько моделей ("комитет") голосуют за спорные примеры.
    - Diversity Sampling (выбор по разнообразию) – отбирает разнообразные примеры, чтобы охватить все аспекты данных.

- Expected Model Change (ожидаемое изменение модели) – выбирает примеры, которые сильнее всего изменят параметры модели.
- Expected Error Reduction (ожидаемое сокращение ошибки) – выбирает примеры, которые максимально уменьшат ошибку модели.
- Density-Weighted Methods (методы с учетом плотности данных) – комбинирует неопределенность и распределение данных.

14 KNN (k-ближайших соседей)

1. **Как работает алгоритм KNN?** Он классифицирует или предсказывает значение объекта на основе k ближайших соседей из обучающей выборки. Для этого он вычисляет расстояния между объектами (например, евклидово) и выбирает наиболее частый класс (классификация) или среднее значение (регрессия) среди ближайших точек. KNN не обучает модель явно, а запоминает всю обучающую выборку.

2. **Какие гиперпараметры есть у KNN (k, метрика расстояния)?**

   - k (количество соседей) – определяет, сколько ближайших точек учитывается при классификации/регрессии. Слишком малое k (например, 1) несёт высокий риск переобучения (чувствительность к шуму). Слишком большое k (например, 50) может приводить к сглаживанию границ (недообучению).
   - Метрика расстояния – определяет, как вычисляется расстояние между объектами:
     - Евклидово расстояние (подходит для непрерывных признаков)
     $$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$
     - Манхэттенское расстояние (устойчивее к выбросам)
     $$d(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$
     - Косинусное расстояние (используется для текстовых данных или высокой размерности)
     $$\text{similarity}(x, y) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$
     - Минковского (обобщение евклидова и манхэттенского (параметр p))
     $$d(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{1/p}$$

- Веса соседей – все соседи имеют одинаковый вес либо же вес обратно пропорционален расстоянию до объекта.

3. **Какие проблемы возникают при использовании KNN и как их решить?** Проблемы с набором возможных решений:
   - Высокая вычислительная сложность (медленная работа на больших данных)
     - Оптимизированные структуры данных (KD-Tree, Ball Tree).
     - Снижение размерности (PCA, t-SNE).
     - Приближенные методы (LSH).
   - Чувствительность к шуму и выбросам (искажение расстояний из-за выбросов)
     - Увеличение k.
     - Взвешенное голосование.
     - Предобработка данных.
   - Необходимость нормировки признаков (доминирование признаков в разных масштабах)
     - Нормировка (StandardScaler, MinMaxScaler).
     - Косинусное расстояние.
   - Проклятие размерности (низкая информативность расстояний в многомерных пространствах)
     - Снижение размерности (PCA, UMAP).
     - Косинусное сходство.
   - Дисбаланс классов (предсказание доминирующего класса)
     - Взвешенное голосование.
     - Балансировка (SMOTE).
   - Выбор метрики расстояния (неправильная метрика снижает точность)
     - Тестирование метрик через кросс-валидацию.
     - TF-IDF + Косинусное сходство для текстов.
   - Оптимальный выбор k (переобучение/недообучение)
     - Подбор k через GridSearchCV.
     - Эвристика $k \approx \sqrt{n}$.

4. **Как работает метод ближайших соседей для регрессии?** Метод ближайших соседей для регрессии предсказывает значение целевой переменной нового объекта, усредняя (или беря медиану) значения его k ближайших соседей из обучающей выборки. Алгоритм вычисляет расстояния между объектами, выбирает k ближайших точек и использует их целевую переменную для предсказания. Ключевые параметры: число соседей (k), метрика расстояния и веса (к примеру, обратные расстояния). Для устойчивости метода данные требуют нормировки.

15 Кластеризация

1. **Назовите методы кластеризации, чем они отличаются.**

   - **K-Means**: делит данные на сферические кластеры с центроидами. Требует задания числа кластеров, чувствителен к шуму.
   - **DBSCAN**: группирует по плотности, работает с кластерами произвольной формы. Автоматически определяет число кластеров и игнорирует шум.
   - **Иерархическая кластеризация**: строит дендрограмму для выбора числа кластеров. Подходит для вложенных структур, но вычислительно затратна.
   - **GMM (Gaussian Mixture Models)**: моделирует данные как смесь гауссиан, поддерживает вероятностную принадлежность к кластерам.
   - **Спектральная кластеризация**: эффективна для нелинейных данных, использует матрицу сходства и спектральные свойства.
   - **Mean Shift**: автоматическое определение кластеров через плотность, настройка радиуса окна.

   Различия:

   - **Форма кластеров**: K-Means – сферические, DBSCAN и Mean Shift – произвольные, GMM – эллиптические.
   - **Устойчивость к шуму**: DBSCAN явно помечает выбросы, Mean Shift игнорирует точки с низкой плотностью.
   - **Число кластеров**: K-Means, GMM, спектральная требуют задания числа, DBSCAN, Mean Shift и иерархическая определяют автоматически.
   - **Сложность**: иерархическая и спектральная – $O(n^3)$, K-Means – $O(n)$, DBSCAN – $O(n \log n)$, Mean Shift – $O(n^2)$.

2. **Какие методы не требуют задания числа кластеров?** DBSCAN, иерархическая кластеризация, Mean Shift. Пояснение: эти методы анализируют структуру данных (плотность, иерархию) для выявления естественного числа кластеров.

3. **Как работает EM-алгоритм в кластеризации?** EM-алгоритм (Expectation-Maximization) используется для поиска параметров вероятностных моделей, таких как GMM, где данные считаются порожденными несколькими распределениями. Он итеративно находит параметры распределений, максимизируя правдоподобие данных:

   - E-шаг: вычисляет вероятности принадлежности точек к кластерам через текущие параметры.

- M-шаг: обновляет параметры (средние, ковариации, веса) кластеров на основе этих вероятностей.

4. **Как работает метод k-средних?** Он группирует данные в k кластерах с центроидами, минимизируя сумму квадратов расстояний от точек до центров кластеров:

   (a) Инициализация: выбираются k случайных центроид.

   (b) Назначение кластеров: каждая точка приписывается к ближайшей центроиде (евклидово расстояние).

   (c) Обновление центроид: новый центроид = среднее всех точек кластера:

   $$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

   (d) Повторение: шаги b-c, пока центроиды не стабилизируются или не достигнут максимум итераций.

5. **Что такое матрица смежности в кластеризации?** Это квадратная матрица, где элемент A_{ij} отражает близость или схожесть (смежность нод) между объектами i и j. Она используется в алгоритмах, основанных на графах (например, спектральная кластеризация), чтобы представить данные как сеть связей.

6. **Как работает метод DBSCAN?** Он группирует данные по плотности, используя параметры ε (радиус окрестности) и $min\_samples$ (минимальное число точек в ε-окрестности).

   - Ядровые точки ($\geq min\_samples$ в ε-окрестности) образуют кластеры.
   - Граничные точки присоединяются к кластерам, если находятся в ε-окрестности ядровых.
   - Шум – точки вне кластеров.

   Алгоритм находит ядровую точку, расширяет кластер, включая все достижимые точки через ε-окрестности, повторяется для необработанных точек.

16 Тематическое моделирование

1. **Что такое тематическое моделирование и зачем оно нужно?** Тематическое моделирование (topic modeling) – это метод автоматического выявления скрытых тем в текстовых данных. Оно помогает структурировать коллекции документов, находить основные темы и анализировать большие текстовые корпуса без ручной разметки.

2. **Как EM-алгоритм используется в тематическом моделировании?** EM-алгоритм (Expectation-Maximization) применяется для оценки скрытых параметров тематической модели, например в LDA (Latent Dirichlet Allocation).

   - E-шаг: рассчитываются вероятности принадлежности слов к темам.
   - M-шаг: обновляются распределения тем по словам и тем по документам.

3. **Примеры регуляризаторов в тематических моделях.** Регуляризаторы помогают сделать темы интерпретируемыми и устойчивыми. Примеры:

   - Sparsity (разреженность): заставляет каждое слово принадлежать к небольшому числу тем.
   - Decorrelator (декоррелятор): снижает схожесть между темами.
   - Smoothness (сглаживание): делает распределения более равномерными.

17 Ранжирование и поиск

1. **Зачем нужны ранжирующие системы?** Ранжирование упорядочивает документы по степени релевантности запросу. Это основа информационного поиска: цель – показать пользователю наиболее полезные и соответствующие документы первыми. Ранжирование также применяется в рекомендательных системах, системах вопрос-ответ и др. задачах.

2. **Как оценивают качество поисковых систем?** Используются специальные метрики ранжирования:

   - Precision@k – точность среди топ-k результатов;
   - Mean Average Precision (MAP) – среднее по всем запросам значение средней точности;
   - NDCG (Normalized Discounted Cumulative Gain) – учитывает порядок и степень релевантности;
   - MRR (Mean Reciprocal Rank) – усреднённый обратный ранг первого релевантного результата.

   Метрики могут быть бинарными (релевантен/нет) и градуированными (степень релевантности).

18 Трансформеры

1. **Что такое attention?** Attention – механизм, позволяющий на каждом шаге обработки выделять наиболее значимые части входа.
   Формально: $attention(q, K, V) = softmax(qK^\intercal/\sqrt{d})V$, где q – запрос, K – ключи, V – значения.
   Это позволяет учитывать весь контекст при генерации или обработке текста. В трансформерах используется **Multi-Head Attention**, где вычисляются несколько attention-механизмов параллельно.

2. **В чем отличие между self-attention и cross-attention?**
   Self-attention применяется внутри одного последовательного ввода: запросы, ключи и значения формируются из одного и того же источника (например, предложения).
   Cross-attention – когда запросы берутся из одного источника (например, декодера), а ключи и значения – из другого (например, энкодера). Это важно при генерации текста, перевода и multi-modal задачах.

3. **В чем отличие между transformer, gpt-like и bert-like моделями?**

   - Transformer – архитектура, включающая энкодер и декодер.
   - GPT (Generative Pretrained Transformer) – использует только декодер, обучается автогенеративно (next token prediction).
   - BERT – только энкодер, обучается маскированным языковым моделированием (masked language modeling).

   GPT хорош в генерации, BERT – в понимании текста.

4. **На какие задачи обучался BERT в базовом варианте?**

   - Masked Language Modeling (MLM) – предсказание случайно замаскированных токенов.
   - Next Sentence Prediction (NSP) — модель определяет, является ли второе предложение логическим продолжением первого.

   Эти задачи позволяют BERT понимать контекст в обе стороны.

19 Другое

1. **Что такое метод главных компонент?** PCA (Principal Component Analysis) – метод линейного понижения размерности. Он ищет ортогональные направления (главные компоненты), на которых проекция данных имеет наибольшую дисперсию. Это делается через разложение ковариационной матрицы на собственные вектора.

2. **Что такое переобучение и как с ним бороться?** Overfitting – ситуация, когда модель хорошо работает на обучающей выборке, но плохо – на тестовой. Причина – модель "запоминает" данные, а не обобщает. Способы борьбы: регуляризация (L1/L2), кросс-валидация, ранняя остановка, аугментация данных, dropout.

3. **В чем разница между параметрами и гиперпараметрами модели?**

   - Параметры – обучаются во время тренировки (например, веса нейронной сети).
   - Гиперпараметры – задаются до обучения (например, learning rate, число слоёв, регуляризация). Их подбирают через перебор или байесовскую оптимизацию.

4. **Что такое кросс-валидация и зачем она нужна?** Кросс-валидация – способ оценки обобщающей способности модели. Самый известный – k-fold: данные делятся на k частей, каждая по очереди становится валидацией, остальные – обучением. Среднее качество даёт устойчивую метрику.

5. **Что такое ROC-кривая и AUC?**

   - ROC (Receiver Operating Characteristic) показывает зависимость TPR от FPR при разных порогах классификации.
   - AUC (Area Under Curve) – площадь под ROC, чем ближе к 1, тем лучше. Это метрика качества бинарных классификаторов.

6. **Объясните разницу между байесовским и частотным подходами в ML.**

   - Частотный подход: вероятность = частота в пределе бесконечных наблюдений. Модель оценивает параметры как единственные (MLE).
   - Байесовский подход: вероятность = степень уверенности. Параметры – случайные величины, используется априорное и апостериорное распределение.

7. **Наивный байесовский классификатор, что это такое?** Простая вероятностная модель на основе формулы Байеса с предположением независимости признаков.
   Работает удивительно хорошо для задач текстовой классификации (спам-фильтры, тональность и др.).

$$P(C \mid x_1, x_2, \ldots, x_n) \propto P(C) \cdot \prod_{i=1}^{n} P(x_i \mid C)$$

8. **Что такое “проклятие размерности”?** С увеличением размерности пространство становится разреженным, метрики расстояний теряют смысл, увеличивается потребность в данных. Алгоритмы становятся менее устойчивыми и требуют больше времени и памяти. Особенно остро – в kNN и решающих деревьях.

9. **Как работает метод t-SNE для визуализации данных, в чем минусы?** t-SNE понижает размерность, моделируя вероятности близости точек и минимизируя дивергенцию (KL) между распределениями в исходном и целевом пространстве.
Плюсы – красивая визуализация кластеров.
Минусы – высокая сложность $O(N^2)$, плохо работает на больших данных, не сохраняет глобальную структуру, трудно интерпретируем.

10. **Что такое обучение без учителя?** Unsupervised learning – обучение на неразмеченных данных. Цели: кластеризация, понижение размерности, поиск аномалий. Методы: K-Means, DBSCAN, PCA, Autoencoders.

11. **Какие задачи решает обучение с частичным привлечением учителя (semi-supervised)?**

    - Распознавание речи и текста: большинство аудиофайлов или текстов не размечены, но можно улучшить модель, используя их в обучении.
    - Компьютерное зрение: классификация изображений, сегментация, обнаружение объектов — semi-supervised подходы позволяют использовать большие наборы неразмеченных изображений.
    - Обнаружение аномалий: часто только малая часть данных помечена как “аномалия”, и semi-supervised подход помогает лучше оценивать нормальность.
    - Снижение переобучения: использование неразмеченных данных как регуляризатор – модель не может “запомнить” все метки, когда их мало, и начинает учиться обобщать.

12. **Что такое Transfer Learning?** Перенос модели, обученной на одной задаче, на другую. Часто используется в NLP и CV: например, BERT обучается на большом корпусе, а потом дообучается на конкретной задаче (классификация, извлечение сущностей и т.д.).

13. **Что такое "обучение с учителем"(supervised learning)?** Supervised learning – обучение на размеченных данных (есть пары «вход — правильный ответ»). Цели: классификация, регрессия, ранжирование. Методы: логистическая регрессия, SVM, деревья решений, нейросети.

14. **Что такое “теория обучения” (bias-variance tradeoff)?** Теория обучения описывает, как модель обобщает на новых данных, включая

компромисс между смещением и дисперсией. Смещение – это систематическая ошибка, дисперсия – чувствительность к обучающей выборке. Хорошая модель должна сбалансировать оба.
При увеличении сложности модели смещение уменьшается, но дисперсия растёт. И наоборот. Задача – найти баланс между ними для минимальной общей ошибки.