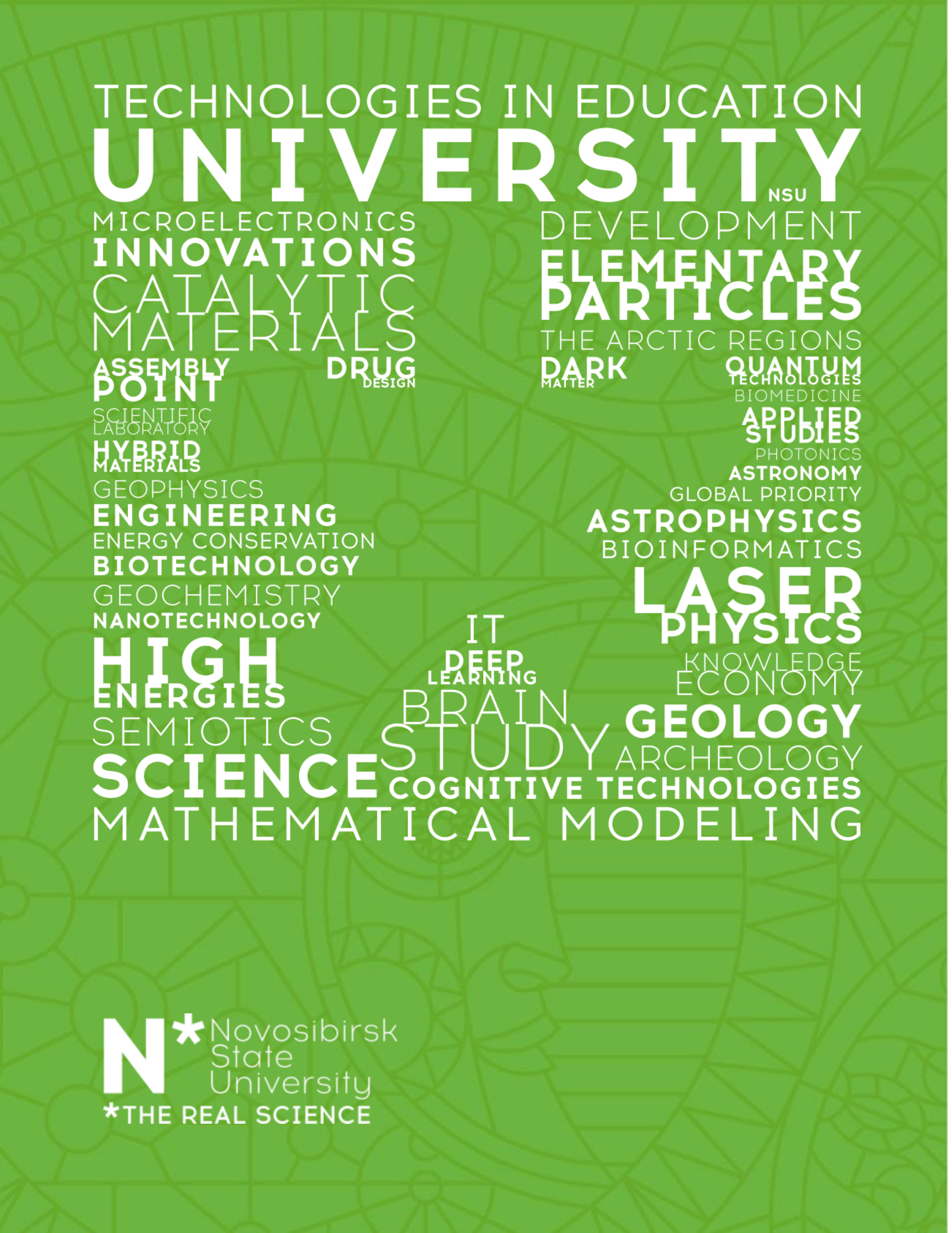


Базовые методы ИИ
Семинар 4

Глушенко Андрей Валерьевич
ФФ НГУ

Практические аспекты

- Оптимальное решение
- Градиентное решение ЛР
- Оптимизаторы градиентного спуска
- L1-, L2-регуляризация
- Теорема Гаусса-Маркова
- Анализ остатков

1

Оптимальное решение

В контексте регрессионного анализа понятие **оптимального решения** — это поиск таких параметров модели, которые минимизируют «ошибку» между предсказаниями алгоритма и реальными данными.

Оптимальное решение

Для поиска оптимального решения нам необходим математический критерий — функция потерь **(Loss Function)**.

Она показывает, насколько сильно мы ошиблись.

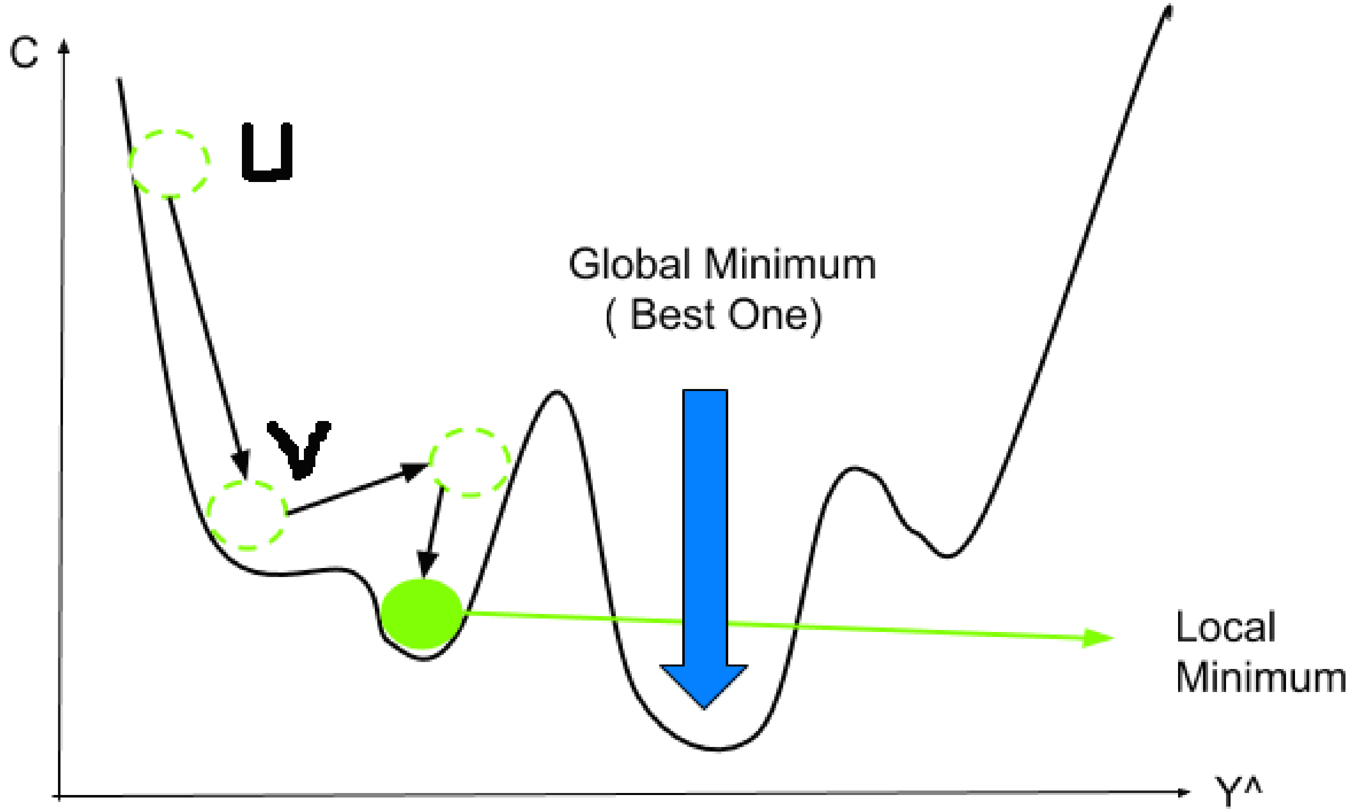
Чаще всего в регрессии используется Метод наименьших квадратов (МНК). В этом случае оптимальным считается решение, при котором сумма квадратов отклонений минимальна:

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \rightarrow \min$$

- y_i — реальное значение.
- $\hat{y}_i$ — предсказанное значение.

Понятие градиентного спуска

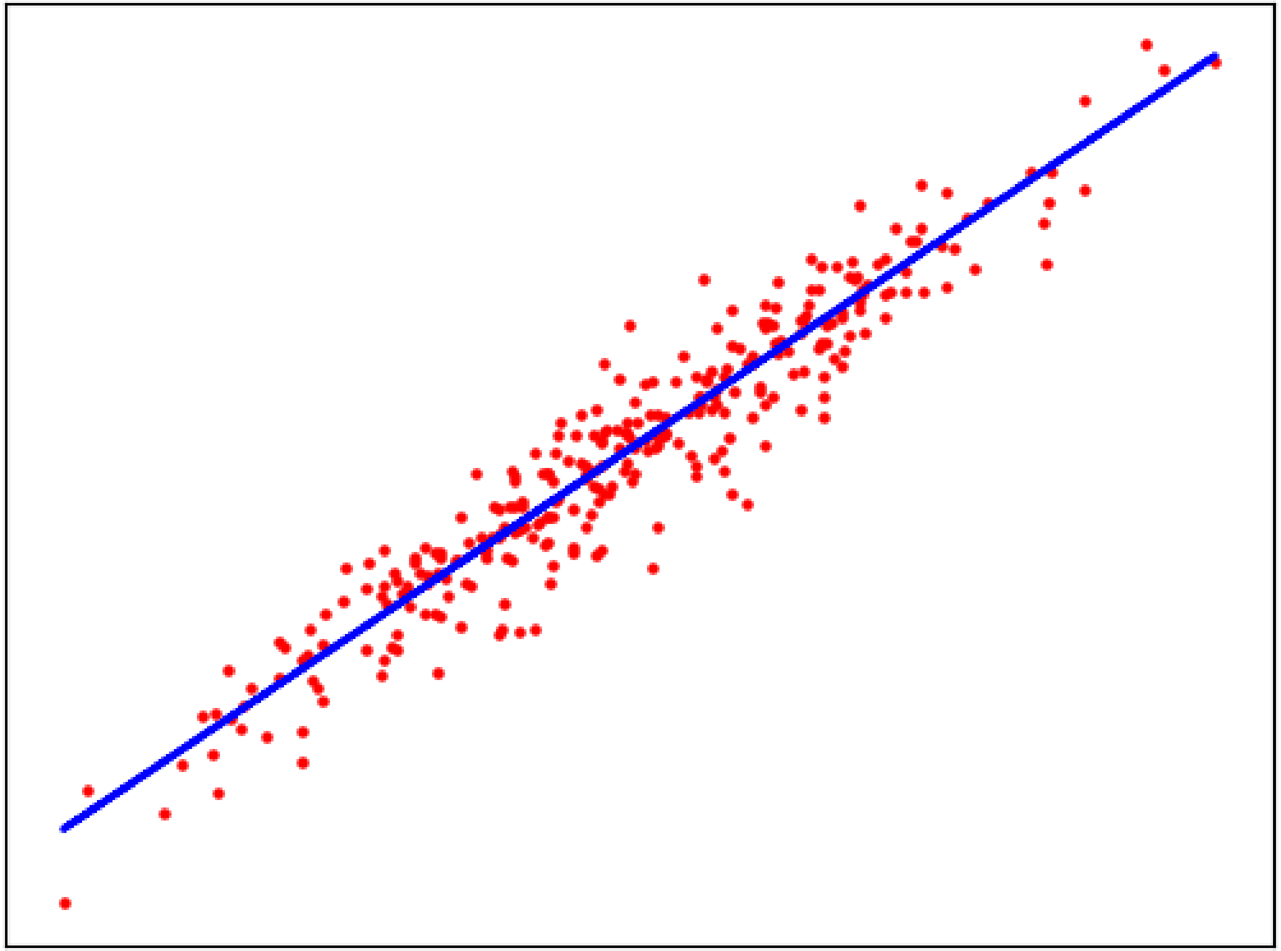
Градиентный спуск — это итерационный алгоритм оптимизации, который используется для поиска минимума функции.



Аналитическое решение ЛР

Имеем решение нормального уравнения:

$$w = (X^T X)^{-1} X^T y$$



- нашли веса за 1 шаг

Вызовы:

- Главная проблема аналитического решения — необходимость инвертирования (обращения) матрицы $X^T X$
  1. Если у нас n признаков (фичей), то сложность обращения достигает $O(n^3)$
  2. Матрица $X^T X$ не всегда обратима. Если признаки сильно коррелируют между собой, аналитическое решение может стать неустойчивым.
- Если набор данных весит **100 ГБ**, а у нас **16 ГБ RAM**, аналитическое решение «вылетит» с ошибкой памяти.

<u>Решение:</u> Изучая градиентный спуск на линейной регрессии, мы осваиваем универсальный «движок», на котором работает почти весь современный Deep Learning.

Градиентное решение ЛР

— это итеративный способ подбора весов модели, при котором параметры постепенно изменяются в сторону уменьшения функции ошибки.

- В отличие от точного аналитического решения, этот метод эффективен на огромных наборах данных

Градиентное решение ЛР. Алгоритм:

1. Инициализация: Веса w и сдвиг b задаются случайными значениями или нулями.
2. Предсказание: $\hat{y} = w \cdot x + b$
3. Расчёт ошибки: $\mathrm{M}SE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 = L$
4. Вычисление градиента:

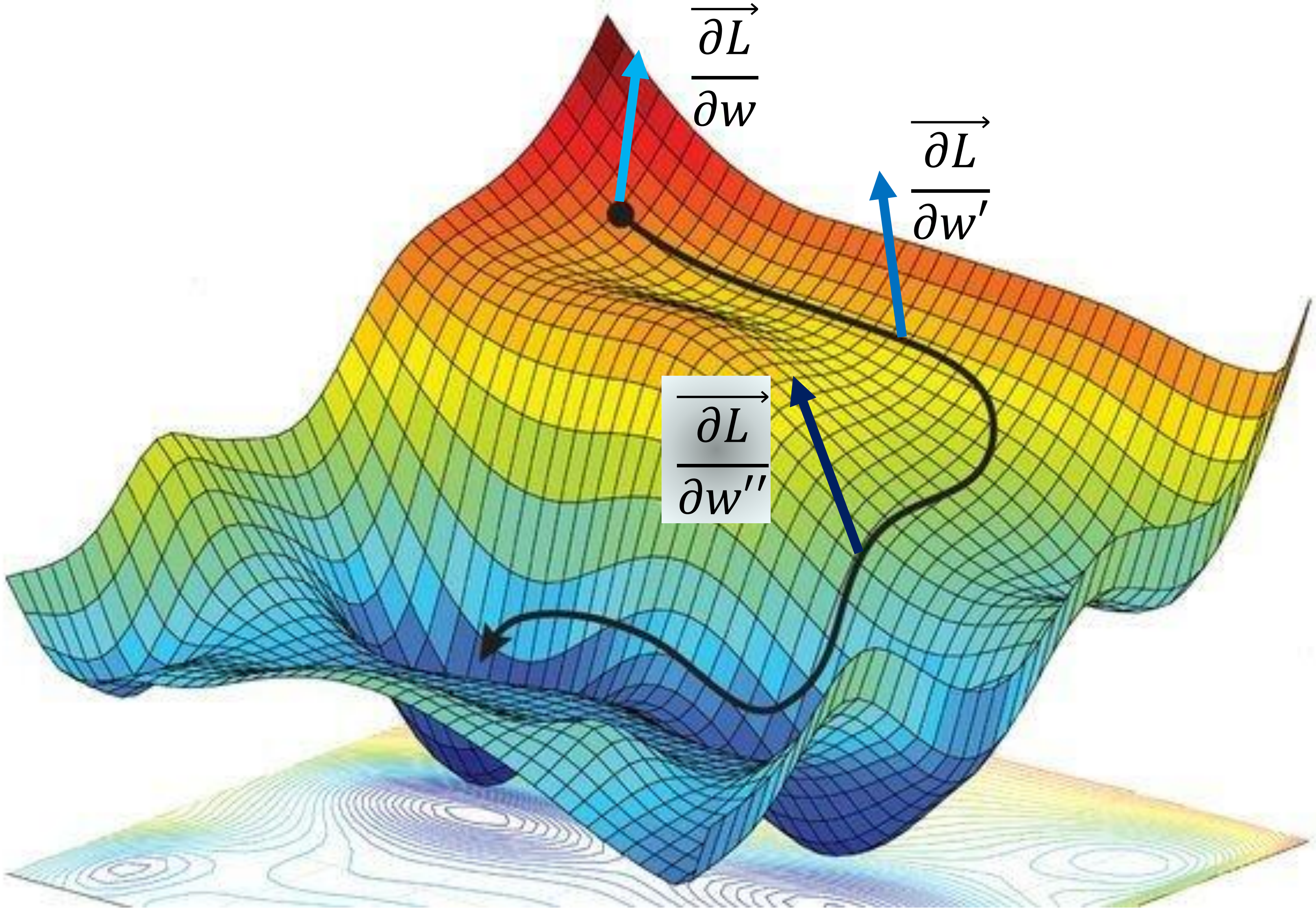
$$\frac{\partial L}{\partial w_j} = \frac{1}{m}\sum_{i=1}^{m} 2(\widehat{y_i} - y_i)\frac{\partial(\widehat{y_i} - y_i)}{\partial w_j} = \frac{2}{m}\sum_{i=1}^{m}(\widehat{y_i} - y_i)x_{ij}\,, \qquad \frac{\partial L}{\partial b} = \frac{2}{m}\sum_{i=1}^{m}(\widehat{y_i} - y_i)$$

5. Обновление весов:

$$w_{new} = w_{old} - \alpha\frac{\partial L}{\partial w_{old}}, \qquad b_{new} = b_{old} - \alpha\frac{\partial L}{\partial b_{old}}$$

α – скорость обучения (гиперпараметр), определяет размер шага

Градиентное решение ЛР



Vanilia SGD (Stochastic Gradient Descent)

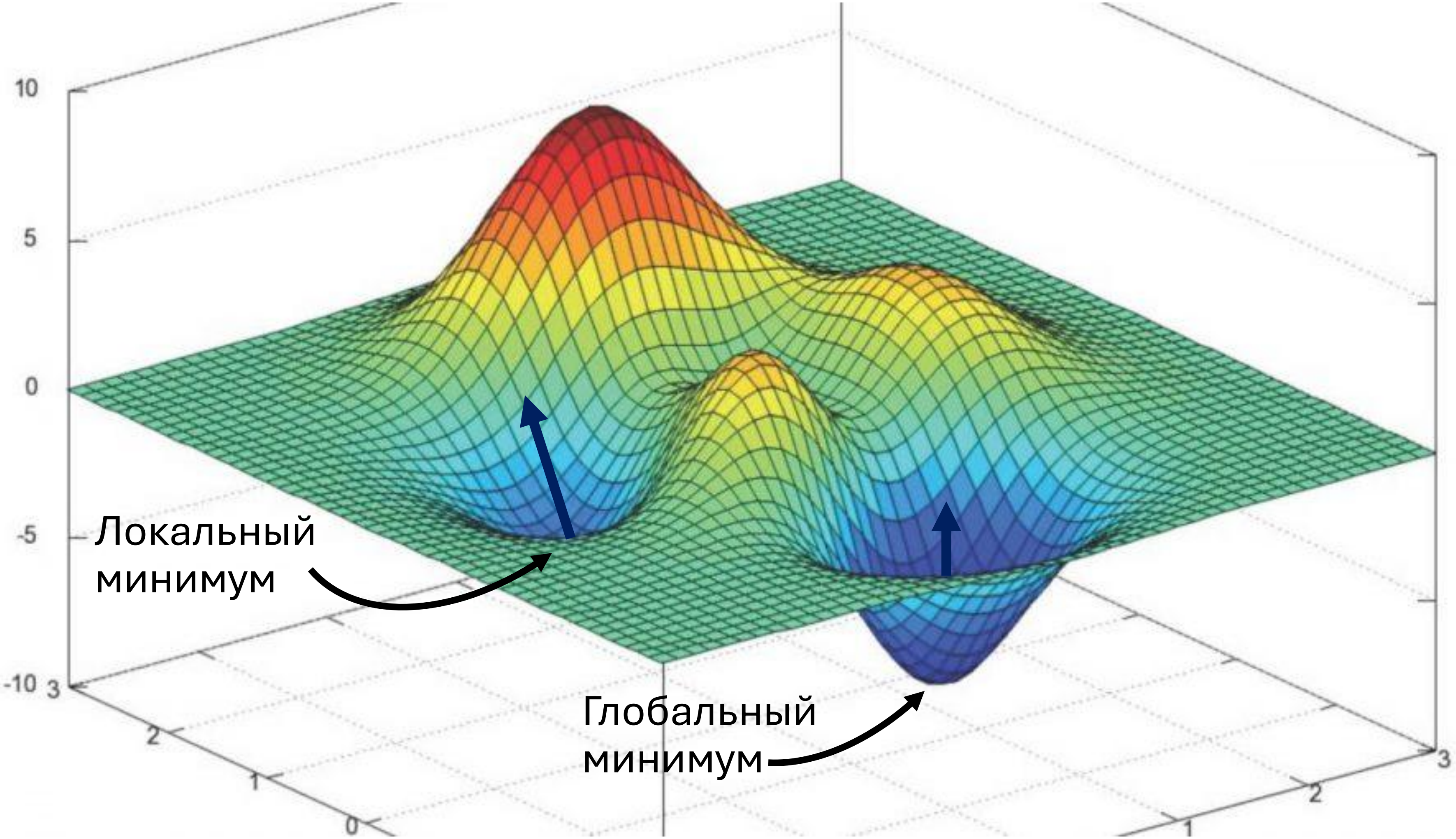
В обычном градиентном спуске мы считаем ошибку по **всем** данным, прежде чем один раз обновить веса. Если данных миллион, это очень долго.

Vanilia SGD:

1. Выбирает **один случайный объект** из обучающей выборки
2. Считает градиент только для этого объекта
3. Мгновенно обновляет веса: $w = w - \alpha\nabla\mathrm{L}$
4. Повторяет это для следующего случайного объекта

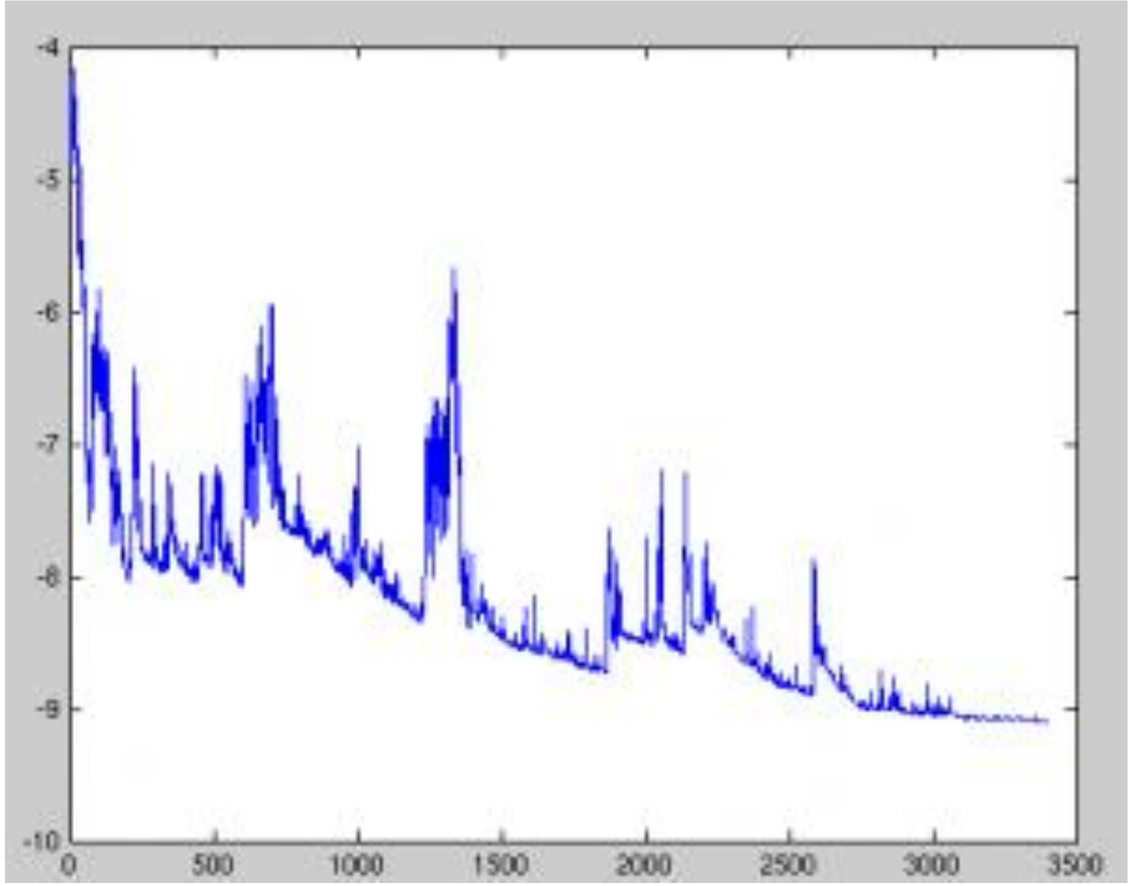
Vanilia SGD (Stochastic Gradient Descent)

- Рандомизация позволяет выйти из локального минимума



Vanilia SGD (Stochastic Gradient Descent)

- Чувствителен к настройке гиперпараметра α
- График зависимости Loss с ростом числа итераций:



Оптимизаторы градиентного спуска

Когда мы переходим от простого градиентного спуска (Vanilla SGD) к реальным задачам, мы сталкиваемся с проблемами:

- алгоритм может застревать в локальных минимумах
- алгоритм может «осциллировать» (сильно прыгать из стороны в сторону)
- алгоритм может обучаться слишком медленно.

Для решения этой проблемы существуют оптимизаторы — надстройки над градиентным спуском, которые адаптивно меняют скорость обучения ($learning\ rate$) α.

Адаптивные оптимизаторы

RMSProp (Root Mean Square Propagation):

- Идея: Алгоритм хранит «бегущее среднее» от квадратов градиентов. Иными словами, благодаря ограничению колебаний не допускается бесконечный рост весов
- $s_t = \beta \cdot s_{t-1} + (1 - \beta){g_t}^2$ – обновление среднего квадрата градиента,
- β – гиперпараметр сглаживания (обычно = 0,9)
- $w_{t+1} = w_t - \frac{\alpha}{\sqrt{s_t + \epsilon}} g_t$
- α – скорость обучения (learning rate)
- ϵ – малое число (10^{-8}) для предотвращения деления на 0

Адаптивные оптимизаторы

AdaDelta (решает проблему AdaGrad, где знаменатель рос бесконечно)

- Проблема: В предшественнике (AdaGrad) скорость обучения со временем падала до нуля, и обучение просто останавливалось.
- Решение: AdaDelta ограничивает окно накопленных градиентов фиксированным размером
- Обенность: В базовой версии AdaDelta даже не нужен начальный learning rate — оптимизатор вычисляет его сам.

Адаптивные оптимизаторы

Adam (Adaptive Moment Estimation)

Король современных оптимизаторов. Он объединяет в себе две идеи:

- Momentum (Инерция): Накапливает «скорость» градиента, чтобы пролетать плоские участки и мелкие ямки.
- RMSProp: Адаптирует шаг для каждого веса отдельно.

Теорема Гаусса-Маркова

— это фундаментальный принцип в статистике, который дает теоретическое обоснование того, почему мы вообще используем метод наименьших квадратов (МНК/OLS) для линейной регрессии.

Теорема Гаусса-Маркова

Условия классической линейной регрессии:

- Линейность параметров: $\hat{y} = w_1 x_1 + w_2 x_2 + w_3 {x_3}^2 \quad \cancel{(+ e^{w_3} x_3)}$
- Полный ранг матрицы признаков (Отсутствие мультиколлинеарности, матрица $X^T X$ обратима)
- Экзогенность (Нулевое матожидание ошибки): ошибки должны быть случайными и не зависеть от того, какие значения принимают x_i
- Гомоскедастичность (Постоянная дисперсия): Если при малых значениях x_i модель ошибается чуть-чуть, а при больших модель имеет огромный разброс, это называется гетероскедастичностью
- Отсутствие автокорреляции: ошибки разных наблюдений не должны зависеть друг от друга

Теорема Гаусса-Маркова

Теорема утверждает, что при соблюдении условий классической линейной регрессии, оценка МНК является **BLUE** (Best Linear Unbiased Estimator).

- Best (Лучшая): Она обладает минимальной дисперсией. Это значит, что оценки весов будут максимально стабильными от выборки к выборке
- Linear (Линейная): Оценка весов представляет собой линейную комбинацию наблюдаемых значений зависимой переменной
- Unbiased (Несмещенная): В среднем оценка весов будет равна их истинному значению (она не систематически «промахивается» мимо истины)
- Estimator (Оценка): Алгоритм вычисления параметров

Теорема Гаусса-Маркова

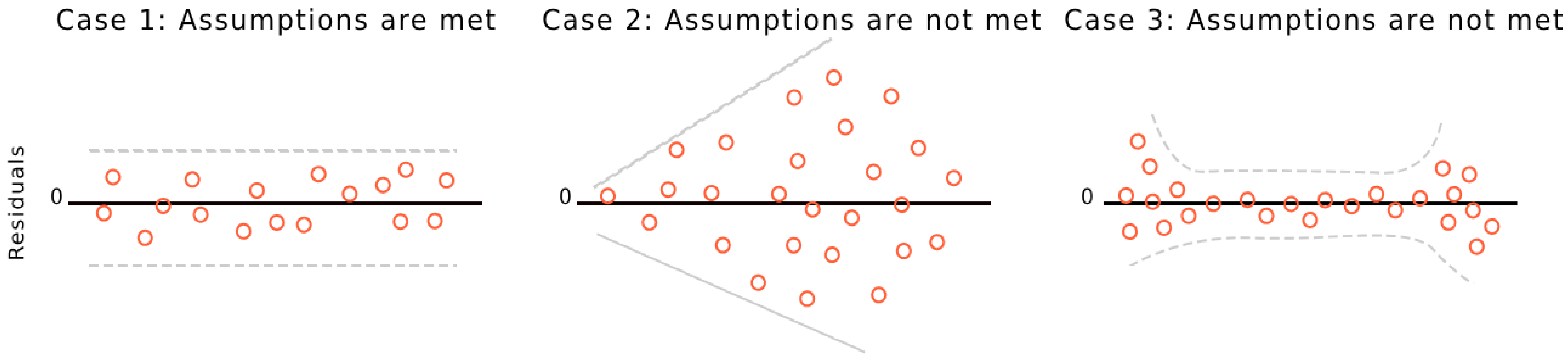
- Теорема не требует, чтобы ошибки были распределены нормально. Она говорит лишь о том, что даже без нормальности МНК — самый эффективный способ поиска весов среди всех линейных методов.

- Градиентный спуск в линейной регрессии сходится к тому же решению, что и аналитический МНК, теорема Гаусса-Маркова гарантирует, что результат будет обладать теми же свойствами BLUE (при соблюдении условий).

Анализ остатков

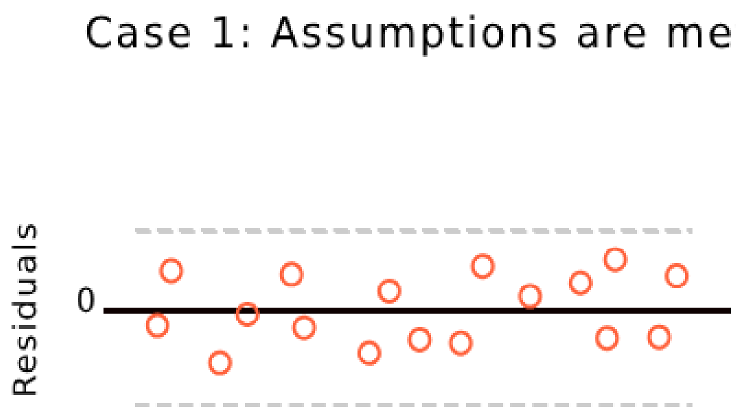
— это «диагностика» обученной модели. Это процесс проверки того, насколько хорошо модель объяснила данные и не остались ли в ошибках какие-то скрытые закономерности

Остатки (residuals): $e_i = y_i - \widehat{y}_i$



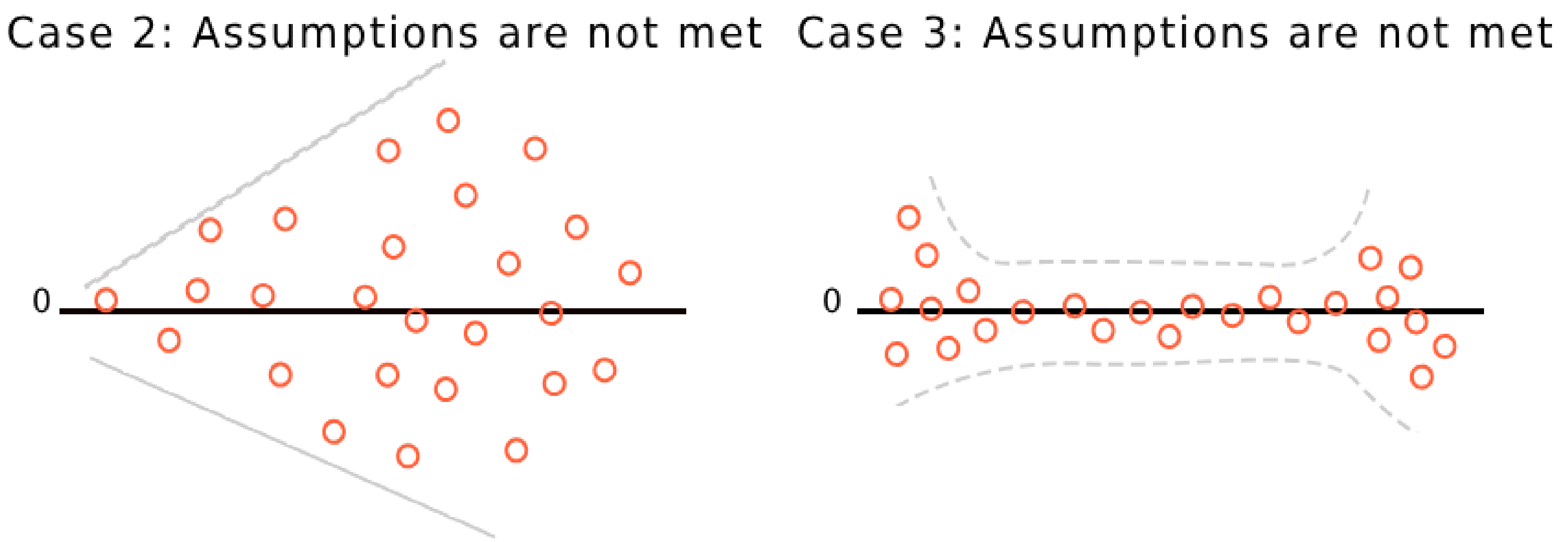
Анализ остатков

- Остатки хаотично разбросаны вокруг нулевой линии.
- Плохо: Остатки образуют дугу, волну или кривую.
- Вывод: Взаимосвязь между x и y на самом деле не линейная (например, квадратичная), нужно добавить полиномиальные признаки x^k



Анализ остатков. Проверка гомоскедастичности (из теоремы Г-М)

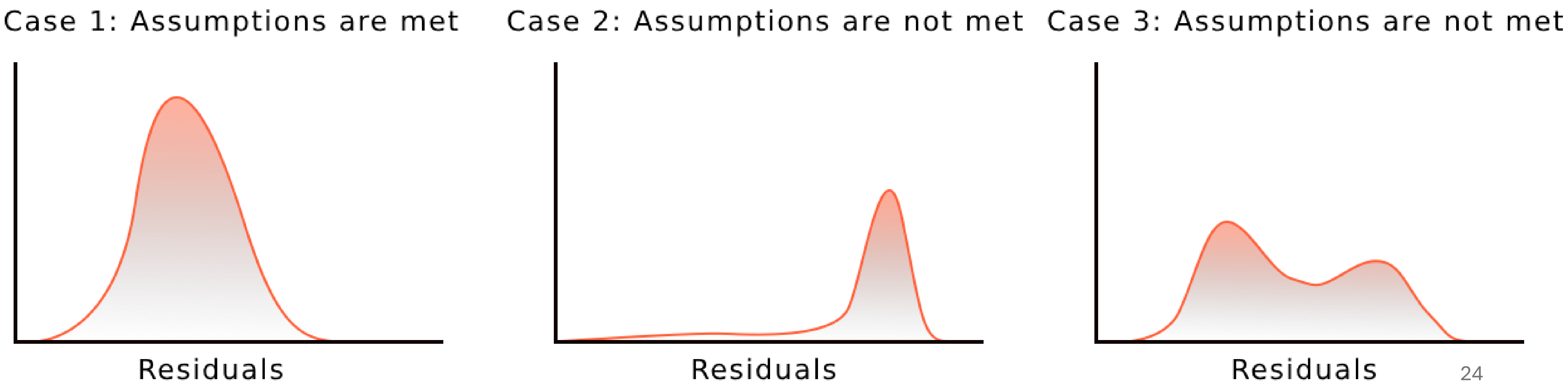
- Хорошо: Облако точек имеет примерно одинаковую ширину (высоту) на всем протяжении.
- Плохо: График напоминает «рупор» (сначала узко, потом широко).
- Вывод: Дисперсия ошибки непостоянна. Это нарушает условия BLUE, и доверие к коэффициентам модели падает.



Анализ остатков. Проверка нормальности распределения

Строим гистограмму остатков или график Q-Q plot.

- Хорошо: Гистограмма напоминает «колокол» (нормальное распределение).
- Вывод: Если остатки распределены нормально, мы можем строить надежные доверительные интервалы и проверять статистическую значимость весов (p-values).



Лабораторная работа

Градиентное решение ЛР

Оптимизаторы градиентного спуска

Теорема Гаусса-Маркова

Анализ остатков