ВНИМАНИЕ!

ДАННЫЙ КУРС СОДЕРЖИТ БОЛЬШОЕ КОЛИЧЕСТВО РАЗНООБРАЗНОГО КОДА И ЗАДАНИЙ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ.

НА ПЕРВЫЙ ВЗГЛЯД ОН МОЖЕТ ПОКАЗАТЬСЯ СЛОЖНЫМ И ТРАВМИРОВАТЬ НЕПОДГОТОВЛЕННУЮ ПСИХИКУ. ТАКЖЕ ОН СОДЕРЖИТ БОЛЬШОЕ КОЛИЧЕСТВО НЕУДАЧНЫХ ШУТОК И НЕУМЕСТНЫХ ОТСЫЛОК.

В СВЯЗИ С ЭТИМ КУРС НЕ РЕКОМЕНДУЕТСЯ ПРОСЛУШИВАТЬ … НИКОМУ.

Машинное обучение 2: Судный день

Ульянкин Филипп

5 сентября 2022 г.

Посиделка 1: вводная с кучей повторения

Agenda

- Бюрократия
- Какие задачи решают в машинном обучении и чем мы займёмся
- Откуда берутся функции потерь и какими они бывают
- Что такое регуляризация
- Что такое BVD

Правила игры

Про пары

- что-то неясно $\Rightarrow$ ПЕРЕБЕЙ И СПРОСИ
- презентаций будет мало, заводите тетрадки :3
- все материалы можно найти на страничке курса

Что почитать

- Онлайн-учебник по машинному обучению от ШАД
- Конспекты Жени Соколова

Карта ML


Машинное обучение
Классическое обучение
Без учителя
Кластеризация
DBSCAN
K-Means
Agglomerative
Mean-Shift
Fuzzy C-Means
Поиск правил
Euclat
Apriori
FP-Growth
Уменьшение размерности (обобщение)
t-SNE
PCA
LSA
SVD
LDA
С учителем
Классификация
Naive Bayes
K-NN
SVM
Decision Trees
Logistic Regression
Регрессия
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
Ансамблевые методы
Стекинг
Беггинг
Random Forest
Бустинг
XGBoost
AdaBoost
LightGBM
CatBoost
Обучение с подкреплением
Генетический Алгоритм
Q-Learning
SARSA
Deep Q-Network (DQN)
A3C
Нейросети и глубокое обучение
Convolutional Neural Networks (CNN)
DCNN
Перцептроны (MLP)
Recurrent Neural Networks (RNN)
LSM
LSTM
GRU
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq

Классическое Обучение

Данные заранее категоризированы или численные

Данные никак не размечены

С учителем

Предсказать категорию

Предсказать значение

Классификация

«Разложи носки по цвету»

Регрессия

«Разложи галстуки по длине»

Без учителя

Разделить по схожести

Найти зависимости

Выявить последовательности

Кластеризация

«Разложи похожие вещи по кучкам»

Ассоциация

«Найди какие вещи я часто ношу вместе»

Уменьшение Размерности (обобщение)

«Собери из вещей лучшие наряды»

Обучение с учителем: регрессия

Базовая постановка

- $(x_i, y_i)_{i=1}^n$ — обучающая выборка
- $a(x)$ — алгоритм, модель
- $Q(a, X)$ — функционал ошибки алгоритма a на выборке X
- Обучение: $a(x) = \arg\min_{a \in A} Q(a, X)$

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, a(x_i))}_{\text{Функция потерь}} + \lambda \cdot \underbrace{R(w)}_{\text{Регуляризатор}} \to \min_w$$

Регрессия

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, a(x_i))}_{\text{Функция потерь}} + \lambda \cdot \underbrace{R(w)}_{\text{Регуляризатор}} \to \min_{w}$$

- Обучение идёт на действительные числа $y_i \in \mathbb{R}$

Регрессия

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, a(x_i))}_{\text{Функция потерь}} + \lambda \cdot \underbrace{R(w)}_{\text{Регуляризатор}} \to \min_{w}$$

- Обучение идёт на действительные числа $y_i \in \mathbb{R}$
- Квадратичное отклонение:

$$L(y, a) = (a - y)^2$$

Регрессия

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, a(x_i))}_{\text{Функция потерь}} + \lambda \cdot \underbrace{R(w)}_{\text{Регуляризатор}} \to \min_{w}$$

- Обучение идёт на действительные числа $y_i \in \mathbb{R}$
- Абсолютное отклонение:

$$L(y, a) = |a - y|$$

Регрессия

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, a(x_i))}_{\text{Функция потерь}} + \lambda \cdot \underbrace{R(w)}_{\text{Регуляризатор}} \to \min_w$$

- Обучение идёт на действительные числа $y_i \in \mathbb{R}$
- Функция потерь Хубера:

$$L_\delta(y, a) = \begin{cases} \frac{1}{2}(y-a)^2, & |y-a| < \delta \\ \delta\left(|y-a| - \frac{1}{2}\delta\right), & |y-a| \geq \delta \end{cases}$$

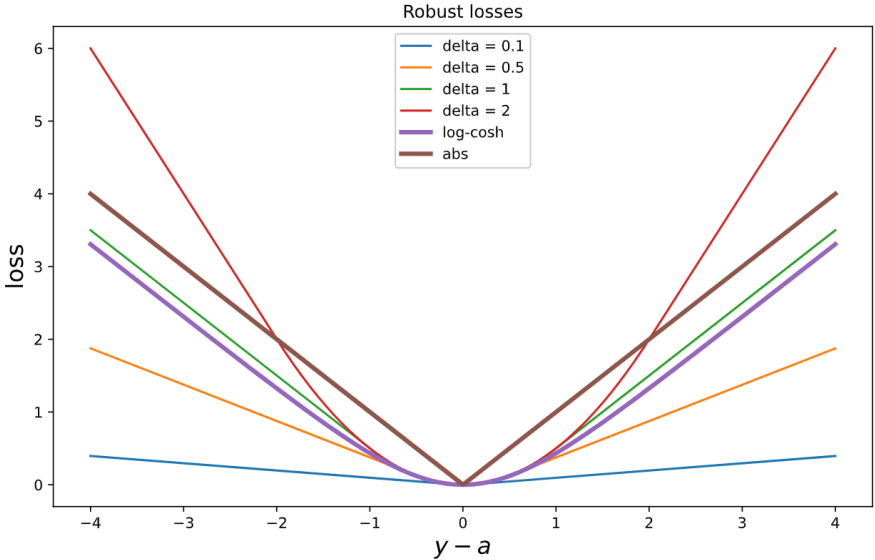
Регрессия

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, a(x_i))}_{\text{Функция потерь}} + \lambda \cdot \underbrace{R(w)}_{\text{Регуляризатор}} \to \min_{w}$$

- Обучение идёт на действительные числа $y_i \in \mathbb{R}$
- Log-Cosh:

$$L(y, a) = \log \cosh(a - y)$$

Функции потерь для регрессии



Инженерный подход к машинному обучению

- Есть проблема, есть функция потерь которая штрафует за ошибку
- Если у этой функции потерь есть какие-то проблемы, стараемся придумать костыли, чтобы их починить
- Квадратичные потери чувствительны к выбросам, абсолютные хуже оптимизируются $\Rightarrow$ скрестим их в потери Хубера
- Вторая производная потерь хубера имеет разрывы $\Rightarrow$ находим похожую гладкую функцию, получаем Log-Cosh
- Статистические свойства не очень нас интересуют

Ещё примеры: Квантильная ошибка

$$L(y, a) = k_1 \cdot |y - a| \cdot [y < a] + k_2 \cdot |y - a| \cdot [y \geq a]$$

- Сложно минимизировать
- В качестве прогноза строится τ-квантиль, где $\tau = \frac{k_2}{k_1 + k_2}$

Ещё примеры: MSLE

$$L(y, a) = (\log(a + 1) - \log(y + 1))^2$$

- Подходит для задач с неотрицательной целевой переменной и неотрицательными прогнозами модели
- За счёт логарифмирования ответов и прогнозов мы скорее штрафуем за отклонения в порядке величин, чем за отклонения в их значениях
- Логарифм не является симметричной функцией, и поэтому данная функция потерь штрафует заниженные прогнозы сильнее, чем завышенные

Вероятностный подход к машинному обучению

- Обучающая выборка и ответы на ней $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ приходят к нам из какого-то распределения

$$p(x, y) = p(x \mid y) \cdot p(y)$$

Вероятностный подход к машинному обучению

- Обучающая выборка и ответы на ней $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ приходят к нам из какого-то распределения

$$p(x, y) = p(x \mid y) \cdot p(y)$$

- Параметризуем это распределение с помощью какой-то модели

$$p(x, y \mid \theta) \propto p(x \mid y, \theta) \cdot p(y)$$

Вероятностный подход к машинному обучению

- Обучающая выборка и ответы на ней $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ приходят к нам из какого-то распределения

$$p(x, y) = p(x \mid y) \cdot p(y)$$

- Параметризуем это распределение с помощью какой-то модели

$$p(x, y \mid \theta) \propto p(x \mid y, \theta) \cdot p(y)$$

- Если предположить, что вектор параметров θ константа, получим метод макимального правдоподобия

$$\hat{\theta} = \arg\max_{\theta} L(\theta) = \arg\max_{\theta} \prod_{i=1}^{n} p(y_i \mid x_i, \theta) = \arg\max_{\theta} \prod_{i=1}^{n} p(x_i \mid dy_i, \theta),$$

Вероятностный подход к машинному обучению

- Обучающая выборка и ответы на ней $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ приходят к нам из какого-то распределения

$$p(x, y) = p(x \mid y) \cdot p(y)$$

- Параметризуем это распределение с помощью какой-то модели

$$p(x, y \mid \theta) \propto p(x \mid y, \theta) \cdot p(y)$$

- Если предположить, что вектор параметров θ случайная величина, мы приоткроем для себя дверь в Байесовские методы

Среднеквадратичная ошибка

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(a(x_i) - y_i)^2$$

- Легко минимизировать
- Чувствительна к выбросам
- В качестве прогноза строится $\mathbb{E}(y \mid X)$
- Обучение эквивалентно оптимизации правдоподобия для

$$y_i \mid x_i \sim N(a(x_i), \sigma^2)$$

- На семинарах будем решать такие задачи :)

Резюме

- Есть два подхода к придумыванию функций потерь: инженерный и вероятностный
- Очень часто функции потерь — замаскированное правдоподобие
- Если уметь сопоставлять функцию потерь вероятностной модели, можно лучше понимать зону её применимости
- Подходы можно комбинировать

Обучение с учителем: классификация

Классификация

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, a(x_i))}_{\text{Функция потерь}} + \lambda \cdot \underbrace{R(w)}_{\text{Регуляризатор}} \to \min_{w}$$

- Обучение идёт на класс $y_i \in \{-1, 1\}$

Классификация

$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, a(x_i))}_{\text{Функция потерь}} + \lambda \cdot \underbrace{R(w)}_{\text{Регуляризатор}} \to \min_{w}$$

- Обучение идёт на класс $y_i \in \{-1, 1\}$
- Доля неправильных ответов:

$$Q(a, X) = \sum_{i=1}^{n} [a(x_i) \neq y_i]$$

Классификация

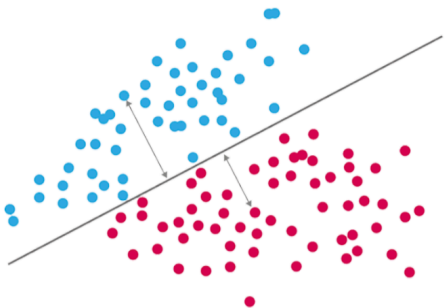
$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, a(x_i))}_{\text{Функция потерь}} + \lambda \cdot \underbrace{R(w)}_{\text{Регуляризатор}} \to \min_w$$

- Обучение идёт на класс $y_i \in \{-1, 1\}$
- Доля неправильных ответов:

$$Q(a, X) = \sum_{i=1}^{n} [a(x_i) \neq y_i] = \sum_{i=1}^{n} [\underbrace{y_i \cdot \langle w, x_i \rangle}_{\text{Отступ}} < 0] = \sum_{i=1}^{n} [M_i < 0]$$

Отступы

- Отступ: $M_i = y_i \cdot \langle w, x_i \rangle$
- Если $M_i > 0$, классификатор даёт верный ответ, если $M_i < 0$, ошибается
- Чем дальше отступ от нуля, тем сильнее классификатор уверен в своей правоте

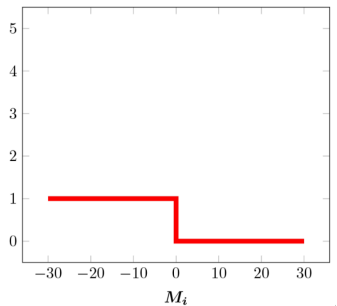


Пороговая функция потерь

- Разрывная функция

- Можно использовать методы негладкой оптимизации, но это сложно

$$L(M_i) = [M_i < 0] = [y_i \cdot \langle w, x_i \rangle < 0]$$



Инженерный подход к классификации

- Возьмём любую гладкую оценку пороговой функции

$$[M < 0] \leq \tilde{L}(M)$$

- Оценим через неё функционал ошибки

$$Q(a, X) \leq \tilde{Q}(a, X) = \frac{1}{n} \sum_{i=1}^{n} \tilde{L}(M_i)$$

- Будем минимизировать получившуюся верхнюю оценку

Примеры оценок

- Логистическая (logloss):

$$\tilde{L}(M_i) = \ln(1 + \exp(-M_i))$$

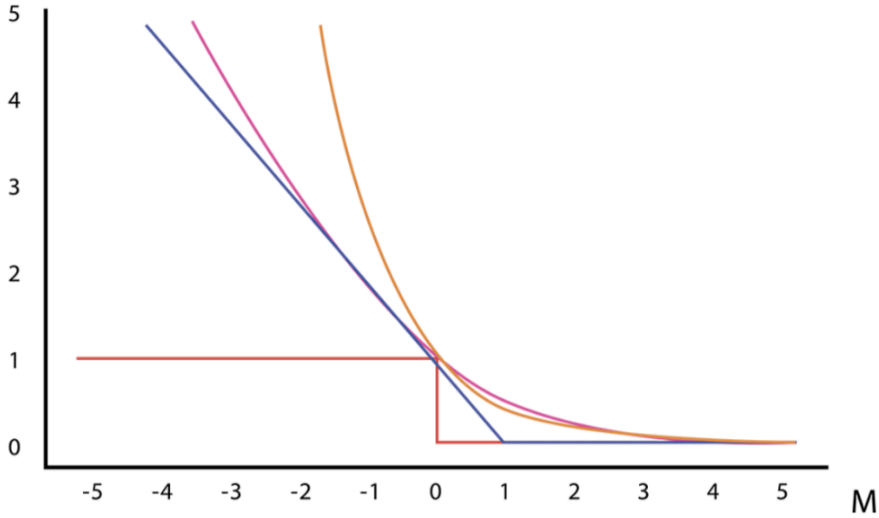
- Экспоненциальная:

$$\tilde{L}(M_i) = \exp(-M_i)$$

- Кусочно-линейная (если добавить L_2 регуляризатор, получим SVM):

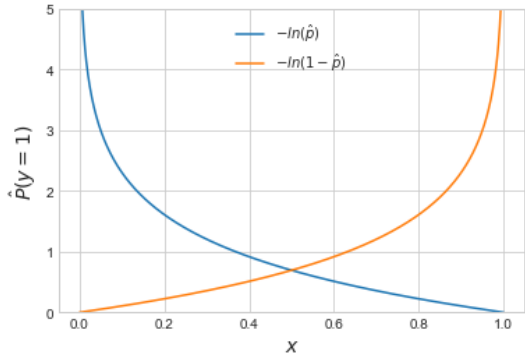
$$\tilde{L}(M_i) = \max(0, 1 - M_i)$$

Примеры оценок



Другой инженерный подход к logloss:

- Пусть y принимают значения 0 и 1
- Если $y = 1$, хотим большое $\hat{p} = \hat{P}(y = 1)$, но чем ближе $\hat{p}$ к 1, тем меньше хотим его увеличить
- Если $y = 0$, хотим большое $(1 - \hat{p})$, получается функция потерь:



$$\text{logloss} = -\frac{1}{n}\sum_{i=1}^{n} y_i \cdot \ln \hat{p}_i + (1-y_i) \cdot \ln(1-\hat{p}_i)$$

Вероятностный подход к logloss

$$L(w) = P(y_1, \dots, y_n \mid X, w) =$$
$$= P(y_1 \mid X, w) \cdot \dots \cdot P(y_n \mid X, w) =$$

Вероятностный подход к logloss

$$
\begin{aligned}
L(w) &= P(y_1, \dots, y_n \mid X, w) = \\
&= P(y_1 \mid X, w) \cdot \dots \cdot P(y_n \mid X, w) =
\end{aligned}
$$

$$
\begin{aligned}
\ln L(w) &= \sum y_i \cdot \ln p_i + \sum (1 - y_i) \cdot \ln(1 - p_i) = \\
&= \sum [y_i \cdot \ln p_i + (1 - y_i) \cdot \ln(1 - p_i)] \to \max_w
\end{aligned}
$$

Вероятностный подход к logloss

$$L(w) = P(y_1, \dots, y_n \mid X, w) =$$
$$= P(y_1 \mid X, w) \cdot \dots \cdot P(y_n \mid X, w) =$$

$$\ln L(w) = \sum y_i \cdot \ln p_i + \sum (1 - y_i) \cdot \ln(1 - p_i) =$$
$$= \sum [y_i \cdot \ln p_i + (1 - y_i) \cdot \ln(1 - p_i)] \to \max_w$$

$$\text{logloss}(w) = -\ln L(w) = -\sum [y_i \cdot \ln p_i + (1 - y_i) \cdot \ln(1 - p_i)] \to \min_w$$

Резюме

- Инженерный подход можно обобщить и для других метрик
- Например, есть гладка верхняя оценка для roc-auc и гладкий аналог f-меры
- На семинарах мы познакомимся с взвешенным logloss и focal-loss

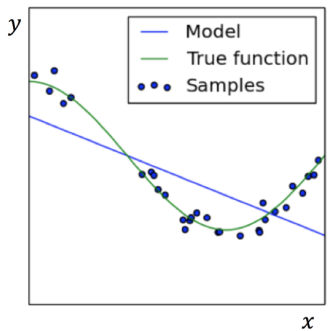
Что такое регуляризация

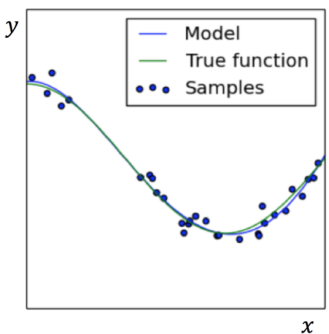
Регуляризация

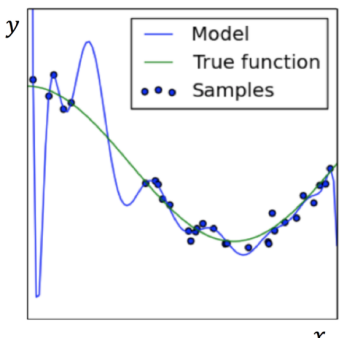
$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, a(x_i))}_{\text{Функция потерь}} + \lambda \cdot \underbrace{R(w)}_{\text{Регуляризатор}} \to \min_{w}$$

- Регуляризация — способ получить решение с определёнными свойствами

Переобучение







Переобучение это

- Модель запомнила данные, излишняя подгонка под обучающую выборку.
- Модель слишком сложная, а данных слишком мало. и она в состоянии их запомнить.
- Признаки: высокое качество модели на обучающей выборке, низкое на тестовой, в случае линейных моделей высокие по модулю коэффициенты.

Чем навеяна регуляризация - сюжет 1

- В хорошей модели: $(0.634, 0.918, -0.626)$
- В переобученной модели: $(130.0, -525.8, \dots, 102.6)$
- Мысль: оштрафовать модель за излишне большие веса, это уберёт изгибы и сделает её проще

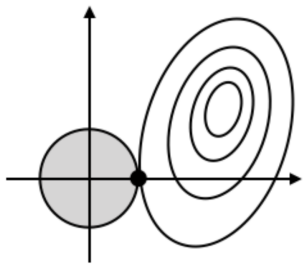
$$Q(a, X) = \frac{1}{n} \sum_{i=1}^{n} \underbrace{L(y_i, a(x_i))}_{\text{Функция потерь}} + \lambda \cdot \underbrace{R(w)}_{\text{Регуляризатор}} \to \min_w$$

L_2 регуляризация (Ridge-регрессия)

$$L(\beta) + \lambda \cdot \sum_{i=1}^{d} \beta_i^2 \to \min_{\beta}$$

Задача оптимизации эквивалентна:

$$\begin{cases} L(w) \to \min_w \\ \sum_{j=1}^{d} w_j^2 \le C \end{cases}$$

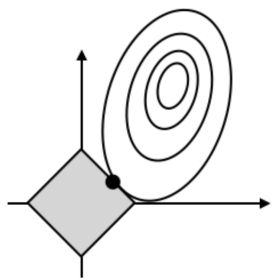


L_1 регуляризация (Lasso-регрессия)

$$L(w) + \lambda \cdot \sum_{j=1}^{d} |w_j| \to \min_w$$

Задача оптимизации эквивалентна:

$$\begin{cases} L(w) \to \min_w \\ \sum_{j=1}^{d} |w_j| \le C \end{cases}$$



Коэффициент регуляризации

- Чем больше λ, тем ниже сложность модели и тем менее сложные закономерности она извлекает из данных;
- Чем меньше λ, тем выше риск переобучения;
- Нужен баланс $\Rightarrow$ подбор λ по кросс-валидации.

Каверзные вопросы

- Вы заметили, что в регуляризатор не включается вес w_0? Почему?

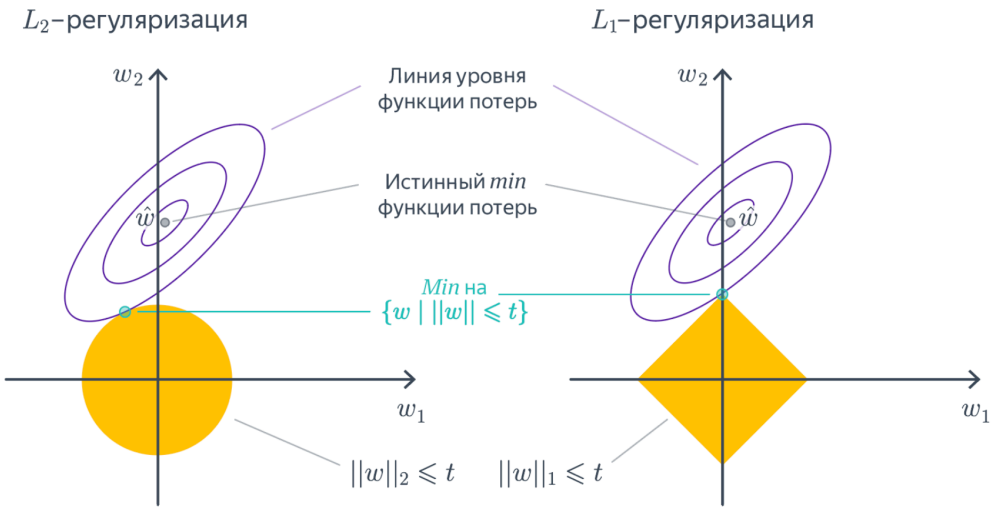
Каверзные вопросы

- Вы заметили, что в регуляризатор не включается вес w_0? Почему?
- Можно ли интерпретировать коэффициенты в Lasso или Ridge регрессии как изменение y в среднем на w_j единиц при росте x_j на единицу?

L_1 vs L_2

- L_2- регуляризатор:
  1. Штрафует модель за сложность
  2. Гладкий и выпуклый
  3. Функция потерь дифференцируема, есть решение в явном виде

- L_1-регуляризатор:
  1. Штрафует модель за сложность
  2. Негладкий
  3. Недифференцируемый, нет решения в явном виде
  4. Некоторые веса оказываются нулевыми, позволяет отбирать признаки



https://ml-handbook.ru/chapters/linear_models/intro

Мультиколлинеарность

- Мультиколлинеарность — линейная зависимость признаков.
- Для любого x_i из выборки существует набор $\alpha_1, \dots, \alpha_d$ такой, что

$$\alpha_1 x_{i1} + \dots + \alpha_d x_{id} = \langle \alpha, x \rangle = 0.$$

Мультиколлинеарность

- Пусть мы нашли решение

$$\hat{w} = \arg\min_{w} \frac{1}{n} \sum_{i=1}^{n} (\langle w, x_i \rangle - y_i)^2$$

- Изменим вектор весов: $\tilde{w} = \hat{w} + t \cdot \alpha$
- Получаем, что

$$\langle \tilde{w}, x \rangle = \langle \hat{w} + t \cdot \alpha, x \rangle = \langle \hat{w}, x \rangle + t \cdot \langle \alpha, x \rangle = \langle \hat{w}, x \rangle$$

- Бесконечно много оптимальных решений

Мультиколлинеарность

- Бесконечно много оптимальных решений
- Большая часть решений с огромными весами $\tilde{w} \Rightarrow$ обладают плохой обобщающей способностью
- Регуляризация, накладывая штраф на коэффициенты, помогает выбрать среди бесконечного числа решений конкретное

Регуляризация и байесовский подход

- На слайдах выше мы относились к регуляризации по-инженерному и ввели её для того, чтобы уменьшить абсолютное значение коэффициентов
- Сущесвтует вероятностный взгляд на регуляризацию
- С точки зрения байесовского подхода, регуляризация соотвествует заданию априорного распределения на коэффициенты
- Подробнее мы поговорим об этом на семинарах

Разложение ошибки на смещение и разброс

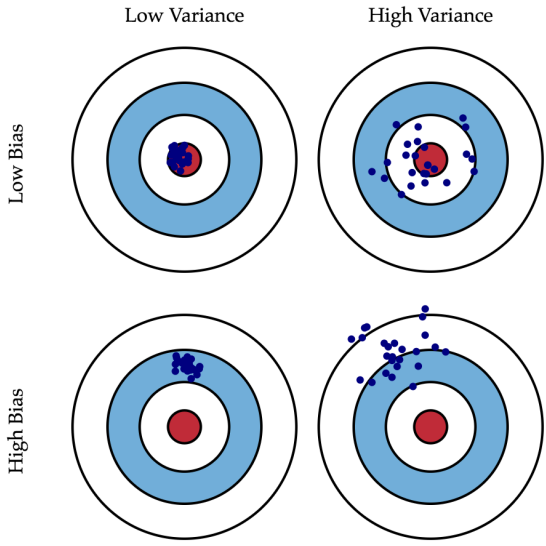
Разложение ошибки на смещение и разброс

- Среднеквадратичную ошибку можно разложить на три составляющие:

$$MSE = \sigma^2 + Var(a(x)) + bias(a(x))^2$$

- σ^2 – неустранимая ошибка;
- $Var(a(x))$ – дисперсия прогноза, то насколько ошибка будет отличаться, если обучать модель на разных наборах данных;
- $bias^2(a(x))$ – средняя ошибка по всевозможным наборам данных;
- С первой ничего сделать не можем, на остальное можем влиять.
- Мы докажем это разложение на следующих лекциях

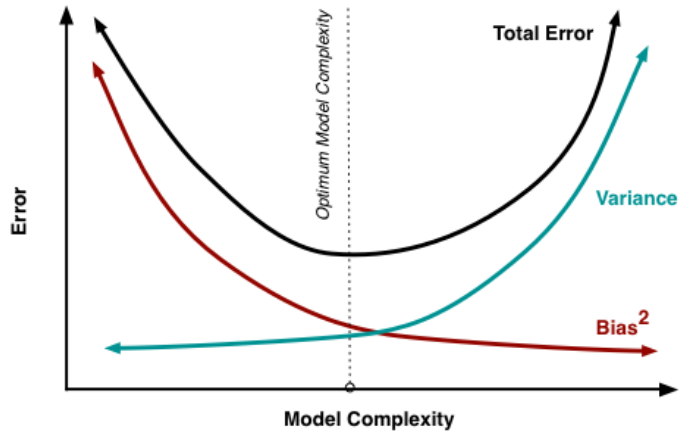
Разложение ошибки на смещение и разброс



Разложение ошибки на смещение и разброс

- При увеличении сложности модели (рост числа оцениваемых параметров) смещение убывает, разброс растёт.
- Модель выучивает тренировочные данные и переобучается.
- Иногда можно намеренно увеличивать смещение модели ради её стабильности.

Разложение ошибки на смещение и разброс



Теорема Гаусса-Маркова

Предположим, что:

1. $y = Xw + \varepsilon$;
2. X — детерминированная матрица размера $n \times k$ с полным столбцовым рангом;
3. $E(\varepsilon) = 0, Var(\varepsilon) = \sigma^2 \cdot I_n$.

Тогда оценка метода наименьших квадратов

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

наиболее эффективная (в смысле наименьшей дисперсии) в классе линейных (по y) несмещённых оценок.

Теорема Гаусса-Маркова и BVD

- Среднеквадратичную ошибку можно разложить на три составляющие:

$$MSE = \sigma^2 + Var(a(x)) + bias(a(x))^2$$

- Теорема Гаусса-Маркова говорит, что $\hat{y}^{OLS}$ обладает нулевым смещением и наименьшим разбросом, то есть для любого другого несмещённого прогноза $\tilde{y}$ выполняется $Var(\tilde{y}) \geq Var(\hat{y})$.

- Прогноз можно сделать смещённым, но с меньшим разбросом с помощью регуляризации

Резюме по регуляризации

- Сюжет 1: регуляризация помогает нащупать баланс между смещением и разбросом и получить более точные прогнозы.

- Сюжет 2: если модель переобучилась, она сложная и извивается. Признак переобучения в линейной модели — большие веса. Регуляризатор вводит штраф за большие веса и даёт бой переобучению.

- Сюжет 3: в случае мультиколлинеарности у нас бесконечно много решений. Многие с большими коэффициентами. Регуляризатор помогает выбрать решение с заданными свойствами: маленькие коэффициенты.

- В общем случае регуляризаторы помогают получать решения с желаемыми свойствами.