Лекція 5

Перенавчання. Метрики якості

§24 Проблема перенавчання

Приклад: проблема перенавчання в задачах класифікації

Припустимо був побудований лінійний класифікатор:

- частка помилок на об'єктах з навчальної вибірки дорівнювала 0.2
- яка частка помилок буде для нової вибірки (0,2; 0,5; 0,9)?

Коли частка помилок **дорівнює 0,9**, то це означає, що алгоритм не зміг узагальнити навчальну вибірку, **не зміг витягнути з неї закономірності** й застосувати їх для класифікації нових об'єктів.

При цьому ***алгоритм якось зміг налаштуватись під навчальну вибірку й показав гарні результати при навчанні без знаходження істинної закономірності***.

У цьому й полягає ***проблема перенавчання***.

Приклад: проблема перенавчання в задачах лінійної регресії

На рисунку зображена істинна залежність та об'єкти навчальної вибірки:

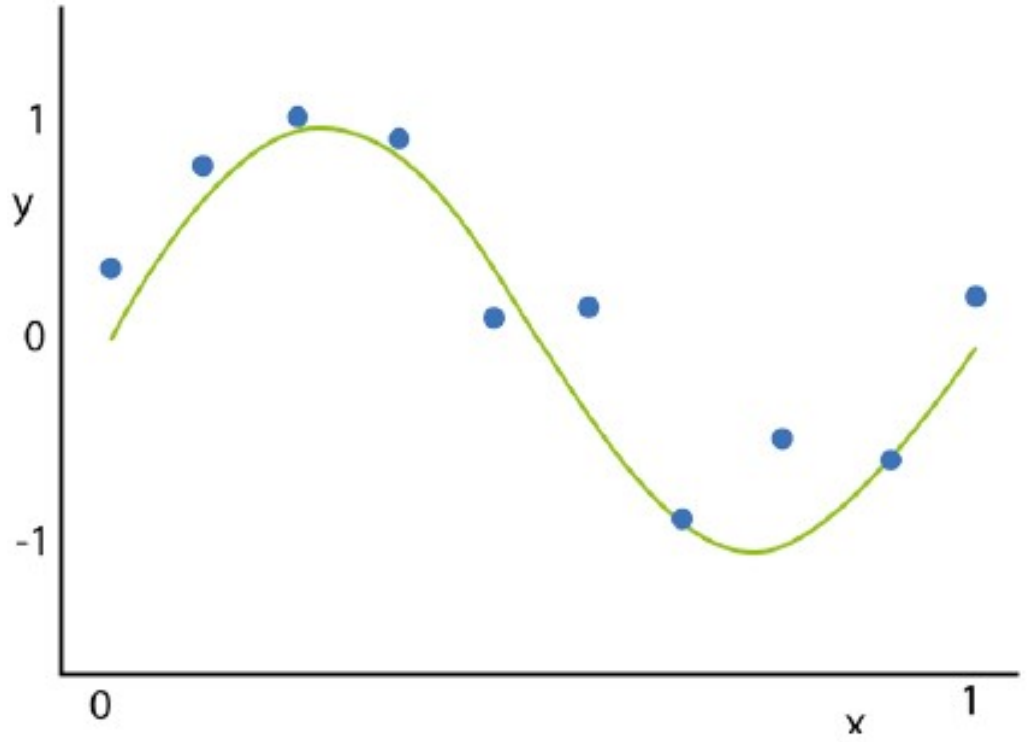


Рис. 3.1: Істинна залежність (зелена лінія) і елементи навчальної вибірки (зображені синіми точками).

Видно, що справжня залежність є нелінійною й має два екстремуми.

У моделі $a(x) = w_0$, після того, як вона буде підігнана під дані, на графіку отримаємо деяку горизонтальну криву.

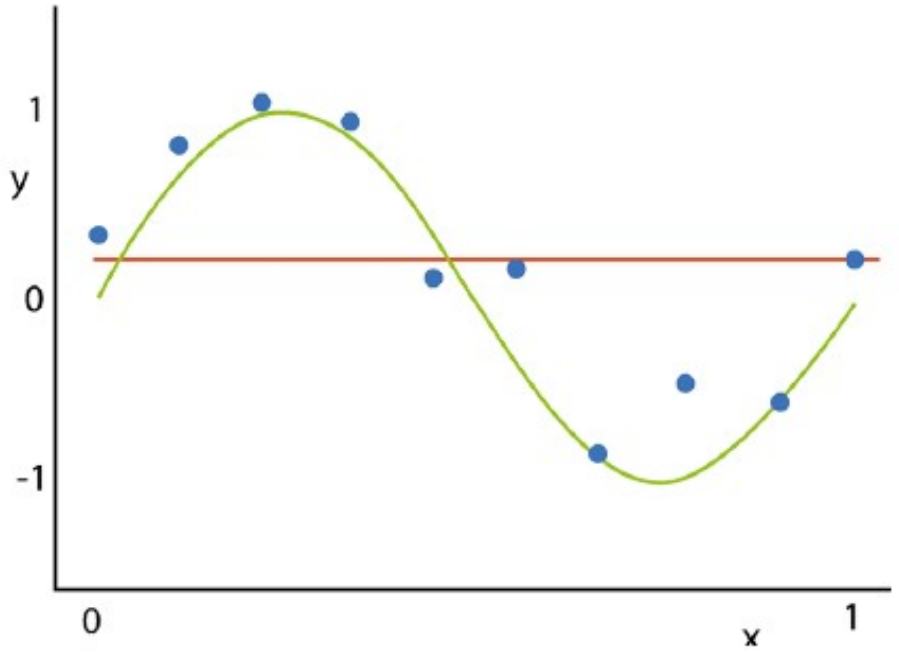


Рис. 3.2: Модель $a(x) = w_0$.

Має місце **недонавчання**.

Гарний алгоритм не був побудований, оскільки сімейство таких алгоритмів не дає змогу вловити закономірність.

Наступна модель $a(x) = w_0 + w_1 x$.

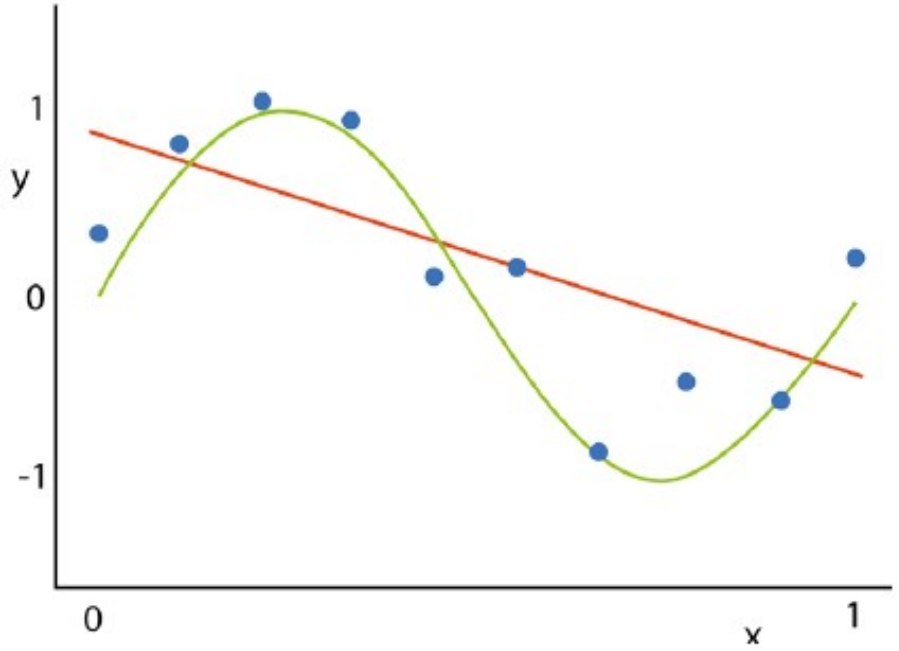


Рис. 3.3: Модель $a(x) = w_0 + w_1 x$.

Має місце **недонавчання**.

Якщо сімейство алгоритмів - множина багаточленів 3-го ступеня:

$$a(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3.$$

Отримана крива буде досить добре описувати й навчальну вибірку, і істину залежність.

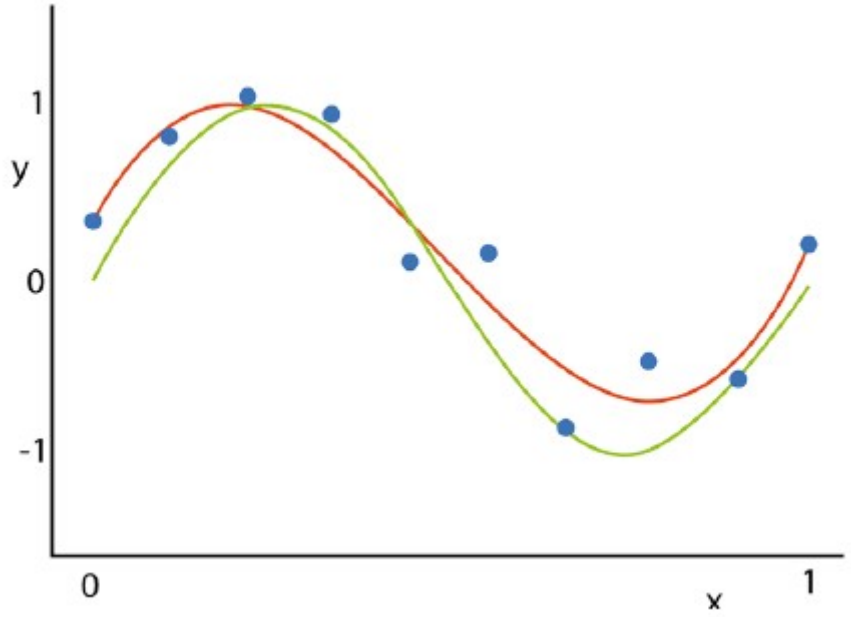


Рис. 3.4: Модель: $a(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$.

У цьому випадку якість алгоритму гарна, але немає ідеального збігу.

Якщо сімейство алгоритмів - множина багаточленів 9-го ступеня:

$$a(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_9 x^9.$$

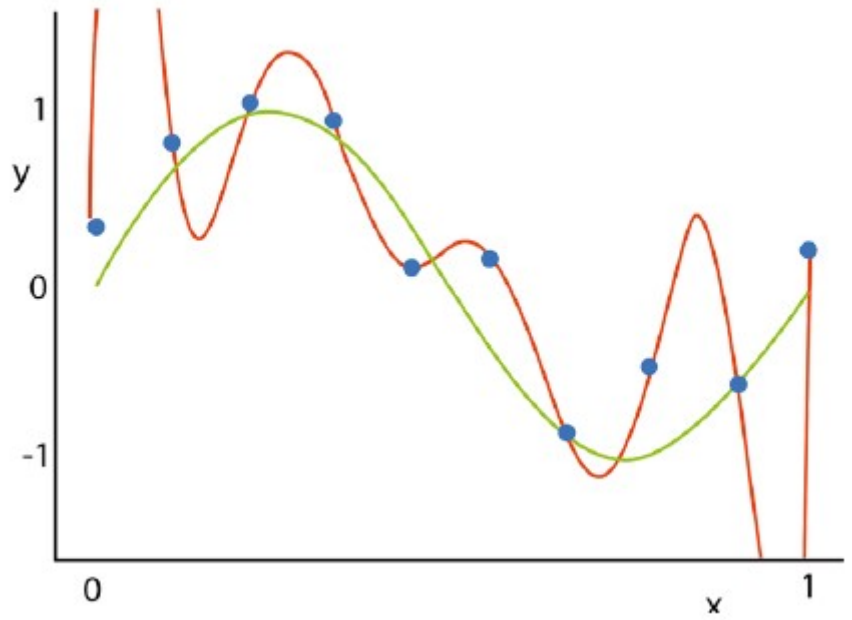


Рис. 3.5: Модель $a(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_9 x^9$.

Відновлена залежність **дає ідеальні відповіді на всіх об'єктах навчальної вибірки**, але при цьому в будь-якій іншій точці **сильно відрізняється від істинної залежності**.

Має місце **перенавчання**.

Недонавчання й перенавчання

- недонавчання – погана якість як на навчанні, так і на нових даних (алгоритм необхідно ускладнювати)
- перенавчання – гарна якість на навчанні, погана на нових даних

Виявити перенавчання, використовуючи тільки навчальну вибірку, неможливо (і добре навчений, і перенавчений алгоритми добре дані описують)

Існують кілька підходів до виявлення перенавчання:

- Відкладена вибірка. Частина даних з навчальної вибірки не беруть участь у навчанні, щоб пізніше перевіряти на ній навчений алгоритм
- Крос-валідація, трохи ускладнений метод відкладеної вибірки
- Використовувати міри якості моделі

§25 Регуляризация

«Симптоми» перенавчання: великі ваги

Мірою складності, тобто «симптомом» перенавченості моделі, є великі ваги біля ознак:

$$a(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_9 x^9$$

ваги виявлялися величезними:

$$a(x) = 0.5 + 12458922x + 43983740x^2 + \ldots + 2740x^9.$$

Порівняйте:

$$a(x) = 0.4 + 8x - 23x^2 + 19x^3$$

Мультиколеніарність

Інша ситуація, з якою можна зустрітися з перенавчанням — **мультиколеніарність** (**лінійна залежність**).

Існують коефіцієнти $\alpha_1, \alpha_2, ..., \alpha_d$ такі, що для будь-якого об'єкта x_i з вибірки виконується:

$$\alpha_1 x^1 + \alpha_2 x^2 + ... + \alpha_d x^d = 0$$

або

$$\langle \alpha, x_i \rangle = 0.$$

Припустимо, було знайдене рішення задачі оптимізації:

$$w_* = \operatorname{argmin}_w \frac{1}{\ell} \sum_{i=1}^{\ell} \left(\langle w, x_i \rangle - y_i \right)^2 .$$

Для іншого вектора ваг, який отриманий переміщенням у напрямку вектора α:

$$w_1 = w_* + t\alpha,$$

також буде рішенням задачі оптимізації, тому що для елементів x вибірки виконується:

$$\langle w_* + t\alpha, x \rangle = \langle w_*, x \rangle + t\langle \alpha, x \rangle = \langle w_*, x \rangle$$

Фактично, вирішеннями задачі оптимізації **є нескінченна множина алгоритмів**, але багато з яких мають великі ваги, і далеко не всі мають гарну узагальнюючу здатність.

Регуляризація

Коли ваги в лінійній моделі великі, існує високий ризик **перенавчання**.

Щоб боротися із цим, мінімізується вже не вираз для функціонала помилки $Q(a, X)$, а новий функціонал, який одержано додаванням регуляризатора.

Найпростіший регуляризатор — квадратичний регуляризатор:

$$\|w\|^2 = \sum_{j=1}^{d} w_j^2.$$

У цьому випадку має місце наступна задача оптимізації:

$$Q(w, X) + \lambda \|w\|^2 \to \min_w.$$

Таким чином, під час навчання буде враховуватися також те, що не варто занадто сильно збільшувати ваги ознак.

Коефіцієнт регуляризації

Уведений вище коефіцієнт λ, що міститься перед регуляризатором, називається коефіцієнтом регуляризації.

- Чим більше λ, тим нижче складність моделі (при дуже великих його значеннях оптимально просто занулити всі ваги).
- При занадто низьких значеннях λ високий ризик перенавчання.

Тому потрібно знайти деяке **оптимальне значення** λ

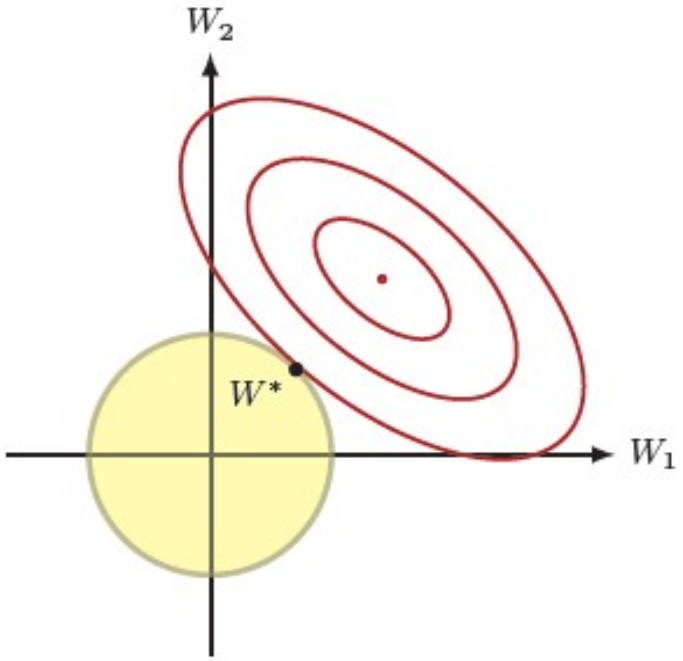
Зміст регуляризації

Розглянемо задачу умовної оптимізації:

$$\begin{cases} Q(w, X) \to \min\limits_{w} \\ \|w\|^2 \leq C \end{cases}$$

Додавання регуляризатора вводить вимогу, щоб рішення задачі мінімізації шукалося в деякій круглій області із центром у нулі.

Рис. 3.6: Геометричний зміст умовної регуляризації. Червона точка — справжній оптимум функції $Q(w, X)$, червоні лінії - лінії рівня функції $Q(w, X)$, чорна точка - оптимум функції при уведеному обмеженні $Q(w, X) + \| w \|^2$.



Види регуляризаторів

Розглянутий вище квадратичний регуляризатор (L_2)

$$\|w\|^2 = \sum_{j=1}^{d} w_j^2.$$

є гладким і опуклим, що дозволяє використовувати градієнтний спуск.

Також існує L_1 регуляризатор:

$$\|w\|_1 = \sum_{j=1}^{d} |w_j|,$$

який є L_1-нормою вектора вагів.

Він уже не є гладким, але має цікаву властивість. Якщо застосовувати такий регуляризатор, **деякі ваги виявляються такими що дорівнюють нулю**. Інакше кажучи, такий регуляризатор виконує відбір ознак і дозволяє використовувати в моделі не всі ознаки, а тільки **найважливіші з них**.

§26 Оцінка якості алгоритмів. Крос-валідація

Виявлення перенавчання

Перенавчання складно виявити, використовуючи тільки навчальну вибірку (і гарний, і перенавчений алгоритми будуть показувати добру якість на об'єктах навчальної вибірки).

Відкладена вибірка

Найпростіший спосіб оцінити якість алгоритму - **використання відкладеної вибірки**.

У цьому випадку варто розбити вибірку на **дві частини**: перша із двох частин буде використовуватися **для навчання** алгоритму, а друга, тестова вибірка, – **для оцінки його якості** (знаходження частки помилок у задачі класифікації, MSE (середньоквадратичної помилки) у задачі регресії)

У **якій пропорції** робити розбивку?

Як правило вибірку розбивають у співвідношеннях 70/30, 80/20 або 0.632/0.368.

Перевагою відкладеної вибірки є те, що навчати алгоритм доводиться всього лише один раз, але при цьому **результат сильно залежить від того, як була зроблена розбивка**.

Наприклад, **оцінюється вартість житла** за деякими ознаками. І є особлива категорія житла, наприклад двоповерхові квартири. І якщо виявиться, що всі двоповерхові квартири, яких небагато, потрапили у відкладену вибірку, то після навчання алгоритм буде давати на них дуже погану якість, оскільки в навчальній вибірці таких об'єктів не було.

Тоді можна використовувати наступний підхід: побудувати n **різних розбивок вибірки** на **2 частині**, для кожної розбивки **знайти оцінку якості**, а як підсумкова оцінка якості роботи алгоритму використовувати усереднене за всіма розбивками значення.

Крос-валідація

Більше системний підхід — **крос валідація**. У цьому випадку вибірка ділиться на ***k* блоків приблизно однакового розміру**. Далі **за чергою кожний із цих блоків використовується в якості тестового, а всі інші - як навчальна вибірка**.

Після того, як кожний блок побуває як тестовий, будуть отримані k показників якості. У результаті усереднення виходить оцінка якості за крос-валідацією.

Яке число блоків використовувати?

Зазвичай вибирають k = 3, 5, 10. Чим більше k, тим більше раз доводиться навчати алгоритм.

Порада: перемішуйте дані у вибірці

Часто дані у файлі записані у відсортованому вигляді за якою-небудь ознакою. Тому завжди варто перемішувати вибірку перш, ніж робити крос-валідацію.

Однак є задачі, у яких вибірку не можна перемішувати. Це задачі передбачення майбутнього, наприклад прогнозу погоди наступного дня. У цьому випадку потрібно особливо стежити за тим, як відбувається ділення вибірки.

§27 Вибір гіперпараметрів і порівняння алгоритмів

Гіперпараметри

Гіперпараметрами називаються **такі параметри** алгоритмів, які **не можуть бути отримані з навчальної вибірки під час навчання**

Прикладами гіперпараметрів є:

- Параметр регуляризації λ (при використанні регуляризатора)
- Ступінь полінома в задачі регресії із сімейством алгоритмів, заданим множиною поліномів відповідного ступеня.

Порівняння різних алгоритмів

Більше загальна задача — порівняння різних алгоритмів:

- навчених з різними значеннями гіперпараметрів;
- які використовують різні способи регуляризації;
- налаштованих з використанням різного функціонала помилки, наприклад середньоквадратичної помилки й середньої абсолютної помилки;
- які належать різним класам алгоритмів.

При порівнянні алгоритмів можна використовувати як відкладену вибірку, так і крос-валідацію, але при цьому слід дотримуватися обережності.

Відкладена вибірка може стати навчальною і виникає проблема перенавчання: з великої кількості алгоритмів вибирається той, який найкраще веде себе на відкладеній вибірці, краще підігнаний під неї.

Поліпшена схема порівняння алгоритмів

- Налаштовуємо усі алгоритми на навчальній вибоці
- Вибираємо кращий на валідаційній виборці
- Найкращий перевіряємо на контрольній виборці

- Налаштовуємо усі алгоритми та вибираємо найкращий на крос-валідації
- Найкращий перевіряємо на контрольній виборці

§28 Метрики якості в задачах регресії

Застосування метрик якості в машинному навчанні

Метрики якості можуть використовуватися:

- Для знаходження функціонала похибки (використовується при навчанні).
- Для підбору гіперпараметрів (використовується при вимірі якості на крос-валідації).
- Для оцінювання підсумкової моделі: чи придатна модель для вирішення задачі.

Середньоквадратична похибка

Перша метрика, про яку вже йшла мова - середньоквадратична похибка:

$$MSE(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} (a(x_i) - y_i)^2 .$$

Такий функціонал **легко оптимізувати**, використовуючи, наприклад, метод градієнтного спуску.

Цей функціонал **сильно штрафує за великі похибки**, тому що відхилення підводяться у квадрат. Це призводить до того, що штраф на викиді буде дуже сильним, і **алгоритм буде підлаштовуватись під викиди.**

Середня абсолютна похибка

Схожий на попереднього функціонала якості - середня абсолютна похибка:

$$MAE(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} |a(x_i) - y_i| .$$

Цей функціонал **складніше мінімізувати**, тому що в модуля похідна не існує в нулі.

Але в такого функціонала **більша стійкість до викидів**, тому що штраф за сильне відхилення набагато менше.

Коефіцієнт детермінації

Коефіцієнт детермінації $R^2(a, X)$:

$$R^2(a, X) = 1 - \frac{\sum_{i=1}^{\ell} (a(x_i) - y_i)^2}{\sum_{i=1}^{\ell} (y_i - \bar{y})^2}, \qquad \bar{y} = \frac{1}{\ell} \sum_{i=1}^{\ell} y_i,$$

дозволяє інтерпретувати значення середньоквадратичної похибки. Цей коефіцієнт **показує, наскільки відрізняються передбачені відповіді** $a(x)$ **відносно істиних** y **по відношенню до розкиду істенних відповідей**.

- R^2 = 1 відповідає випадку ідеальної моделі,
- R^2 = 0 — моделі на рівні оптимальної «константної»
- R^2 < 0 — моделі гірші за «константні» (такі алгоритми ніколи не потрібно розглядати).

Оптимальним константым алгоритмом називається такий алгоритм, що повертає завжди середнє значення відповідей $\bar{y}$ для об'єктів навчальної вибірки.

Несиметричні втрати

До цього розглядалися **симетричні** моделі, тобто такі, які **однаково штрафують** як за недопрогноз, так і за перепрогноз. Але існують такі задачі, у яких ці похибки мають різну ціну.

Приклад:

Нехай, наприклад, потрібно **оцінити попит на ноутбуки**.

- занижений прогноз приведе до **втрати лояльності покупців і потенційного прибутку** (буде закуплена недостатня кількість ноутбуків)
- завищений – тільки до не дуже великих додаткових витрат на зберігання непроданих ноутбуків.

Щоб урахувати це, функція втрат повинна бути **несиметричною** й **сильніше штрафувати за недопрогноз, чим за перепрогноз**.

Квантильна похибка

У таких випадках добре підходить квантильна похибка або квантильна функція втрат:

$$\rho_\tau(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} ((\tau - 1)[y_i < a(x_i)] + \\ + \tau[y_i \geq a(x_i)])(y_i - a(x_i))$$

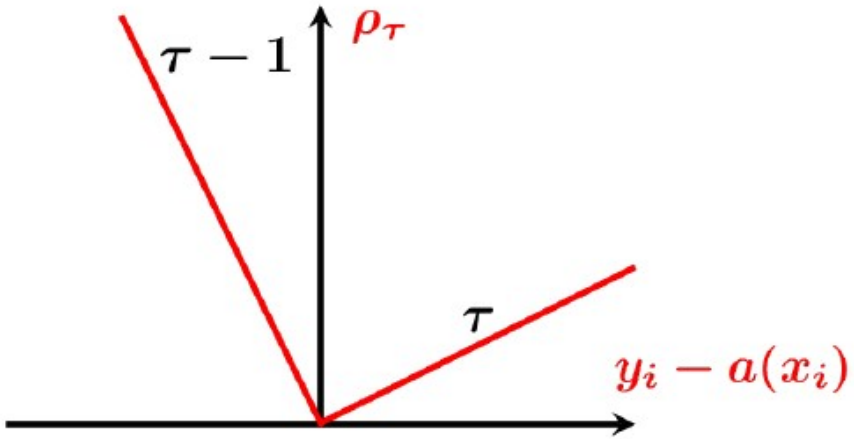


Рис. 4.1: Графік квантильної функції втрат

Параметр $\tau \in [0,1]$ визначає те, за що потрібно штрафувати сильніше — за недопрогноз або перепрогноз. Якщо τ ближче до 1, штраф буде більше за недопрогноз, а якщо, навпаки, ближче до 0 — за перепрогноз.

Імовірнісний зміст похибок

Чому така функція втрат називається **квантильною**?

- Якщо використовується квадратичний функціонал похибки, то найбільш оптимальним прогнозом буде середня відповідь на об'єктах,
- Якщо абсолютний, то медіана відповідей.
- Якщо буде використовуватися квантильна функція втрат, найбільш оптимальним прогнозом, буде τ -квантиль.

τ -квантиль – це число x_τ, яке розділяє дані X на дві частини $X_1 = \{x \mid x \le x_\tau\}$ та $X_2 = \{x \mid x > x_\tau\}$. При цьому частка даних, яка попадає в множину X_1 дорівнює $\tau = \mid X_1 \mid / \mid X \mid$

0.5 -квантиль є медіаною.

τ -квантиль визначає x_τ і множину данних $X_1 = \{x \mid x \le x_\tau\}$, імовірність спостерігання яких дорівнює $\tau = P(x \le x_\tau)$

§29 Метрика якості класифікації

Частка правильних відповідей

Як міру якості в задачах класифікації природно використовувати частку неправильних відповідей:

$$\frac{1}{\ell}\sum_{i=1}^{\ell}[a(x_i) \neq y_i]$$

Однак у **задачах класифікації** прийнято вибирати метрики таким чином, щоб їх **потрібно було максимізувати**, тоді як у задачах регресії - так, щоб їх потрібно було мінімізувати. Тому визначають:

$$\text{accuracy}(a, X) = \frac{1}{\ell}\sum_{i=1}^{\ell}[a(x_i) = y_i]$$

Ця метрика якості проста й широко використовується, однак має кілька істотних недоліків.

Незбалансовані вибірки

Перша проблема пов'язана з **незбалансованими вибірками**.

Приклад: Нехай у вибірці 1000 об'єктів, з яких 950 відносяться до класу (+1) і 50 — до класу (–1). Розглядається **некорисний** (оскільки не відновлює ніяких закономірностей у даних) **константний класифікатор**, такий що на всіх об'єктах повертає відповідь 1. Але **частка правильних відповідей на цих даних буде дорівнює 0.95**, що дещо забагато для некорисного класифікатора.

Щоб «боротися» із цією проблемою, використовується наступний факт. Нехай q_0 — частка об'єктів самого великого класу, тоді **частка правильних відповідей для розумних алгоритмів** *accuracy* [q_0, 1], а не [1/2, 1], як це можна було б очікувати.

Ціни похибок

Друга проблема із часткою вірних відповідей полягає в тому, що вона ніяк **не враховує різні ціни різних типів похибок**. Тоді як ціни дійсно можуть бути різними.

Приклад: У задачі ухвалення рішення щодо видачі кредиту, порівнюються **дві моделі**.

При використанні **першої моделі** кредит буде виданий **100** клієнтам, **80 з яких його повернуть**.

У **другій моделі**, більше консервативної, кредит був **виданий тільки 50** клієнтам, причому **повернули його в 48 випадках**.

Те, яка із двох моделей краща?

Ціна якої з похибок вище: **не дати кредит клієнтові, що міг би його повернути, або видати кредит клієнтові, що його не поверне**.

Частка вірних відповідей не здатна враховувати ціни різних похибок і тому не може дати відповіді на це питання.

§30 Точність і повнота

Матриця похибок

Зручно класифікувати різні випадки за допомогою **матриці похибок**:

	$y = 1$	$y = -1$
$a(x) = 1$	True Positive (TP)	False Positive (FP)
$a(x) = -1$	False Negative (FN)	True Negative (TN)

Коли алгоритм відносить об'єкт до класу +1, говорять, що ***алгоритм спрацьовує***. Якщо алгоритм спрацював і об'єкт дійсно відноситься до класу +1, має місце **вірне спрацьовування (true positive)**, а якщо об'єкт насправді відноситься до класу –1, має місце **хибне спрацьовування (false positive)**.

Якщо алгоритм дає відповідь –1, говорять, що він ***пропускає об'єкт***. Якщо має місце пропуск об'єкта класу +1, то це **хибний пропуск (false negative)**. Якщо ж алгоритм пропускає об'єкт класу –1, має місце **істинний пропуск (true negative)**.

Точність і повнота

Нехай розглядаються дві моделі $a_1(x)$ і $a_2(x)$. Вибірка складається з 200 об'єктів, з яких 100 відносяться до класу +1 і 100 — до класу −1. Матриці похибок мають вигляд:

	$y=1$	$y=-1$
$a(x)=1$	80	20
$a(x)=-1$	20	80

	$y=1$	$y=-1$
$a(x)=1$	48	2
$a(x)=-1$	52	98

Уведемо дві метрики. Перша метрика, **точність (precision**), показує, ***наскільки можна довіряти класифікатору 1 у випадку спрацьовування***:

$$precision(a,X) = \frac{TP}{TP+FP}.$$

Друга метрика, **повнота (recall),** показує, на якій ***частці істинних об'єктів першого класу алгоритм спрацьовує***:

$$recall(a,X) = \frac{TP}{TP+FN}.$$

	$y=1$	$y=-1$
$a(x)=1$	80	20
$a(x)=-1$	20	80

	$y=1$	$y=-1$
$a(x)=1$	48	2
$a(x)=-1$	52	98

У прикладі точність і повнота першого алгоритму виявляються однаковими:

$$\textit{precision}(a_1, X) = 0.8, \quad \textit{precision}(a_2, X) = 0.96$$

$$\textit{recall}(a_1, X) = 0.8, \quad \textit{recall}(a_2, X) = 0.48$$

Висновок:

Друга модель є дуже точною, але повнота є досить низькою.

Повнота характеризує широту охоплення клієнтів.

Приклади використання точності й повноти

Приклад 1

Нехай у задачі кредитного скорінгу ставиться умова, що невдалих кредитів повинне бути не більше 5%. У такому випадку задача є **задачею максимізації повноти** за умови $\text{precision}(a, X) \geq 0.95$.

Приклад 2

Медична діагностика. Необхідно побудувати модель, що визначає, є чи ні певне захворювання у пацієнта. При цьому потрібно, щоб були виявлені як мінімум 80% пацієнтів, які дійсно мають дане захворювання. Тоді ставлять **задачу максимізації точності** за умови $\text{recall}(a, X) \geq 0.8$.

Розглянемо, як **точність і повнота працюють у випадку незбалансованих вибірок**.

Розглядається вибірка з наступною матрицею похибок:

	$y = 1$	$y = -1$
$a(x) = 1$	10	20
$a(x) = -1$	90	10000

$$\text{accuracy}(a, X) = 0.99$$

$$\text{precision}(a, X) = 0.33$$

$$\text{recall}(a, X) = 0.1$$

Те, що частка вірних відповідей **дорівнює 0.99, ні про що не говорить**: алгоритм однаково **робить 66% хибних спрацьовувань** і **виявляє тільки 10% позитивних випадків**.

Завдяки **введенню точності й повноти стає зрозуміло, що алгоритм потрібно поліпшувати**.