Лекция 2
Регрессия и классификация

Макаренко В.А., Габдуллин Р.А.

МГУ им. М.В. Ломоносова

17 января 2023

Обучение с учителем

X – множество объектов,
Y – множество ответов,
$y : X \to Y$ – неизвестная зависимость.

Дано:
$\{x_1, x_2, \dots, x_\ell\} \subset X$ – обучающая выборка,
$y_i = y(x_i),\ i = 1, \dots, \ell$ – известные ответы.

Найти:
$a : X \to Y$ – решающая функция, приближающая y на всём X.

Описание объектов. Признаки

X – множество объектов,
$f_j : X \to F_j, \quad j = 1, \dots, n$ – признаки объектов (features),

Типы признаков:

Бинарные	Binary	$F_j = \{\text{true, false}\}$
Номинальные	Categorical	F_j – конечное мн-во
Порядковые	Ordinal	F_j – конечное упорядоченное мн-во
Количественные	Numerical	$F_j = \mathbb{R}$

$(f_1(x), f_2(x), \dots, f_n(x))$ – признаковое описание объекта $x \in X$.
Матрица «объекты-признаки» (feature data)

$$F = \|f_j(x_i)\|_{\ell \times n} = \begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix}.$$

Классификация

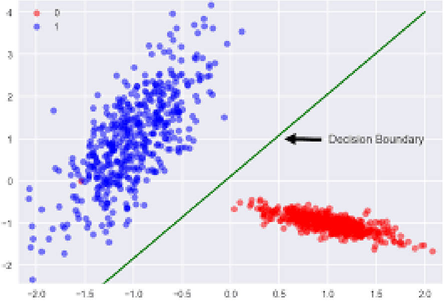


Рис. 1: Источник: kaggle.com

Задача классификации:

- Два класса: $Y = \{0, 1\}$.
- Несколько классов: $Y = \{1, 2, 3, \ldots, m\}$.
- Несколько пересекающихся классов: $Y = \{0, 1\}^m$.

Регрессия

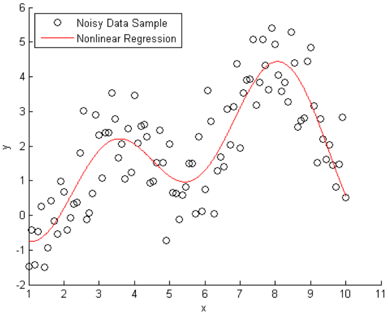


Рис. 2: Источник: datascience.stackexchange.com

Задача восстановления регрессии:

- Вещественный ответ: $Y = \mathbb{R}$ или $Y = \mathbb{R}^m$.

Модель алгоритмов

Зачастую семейство решающих правил (алгоритмов) задается в виде семейства параметрических функций

$$A = \{a(x, \theta) | \theta \in \Theta\},$$

где $a : X \times \Theta \to Y$ – фиксированная функция,
Θ – множество допустимых значений параметра θ.
Пример.
Линейная модель:

$$\theta = (\theta_1, \dots, \theta_n), \quad \Theta = \mathbb{R}^n.$$

Для задачи регрессии:

$$a(x, \theta) = \sum_{k=1}^{n} \theta_k f_k(x).$$

Для задачи классификации:

$$a(x, \theta) = \operatorname{sign} \sum_{k=1}^{n} \theta_k f_k(x).$$

Обучение и применение модели

- Обучение: по объектам и ответам подобрать $\theta \in \Theta$.
  Метод обучения:
  $$\mu : (X, Y)^\ell \to A.$$
  По выборке $\mathbb{X} = (x_i, y_i)_{i=1}^\ell$ получаем алгоритм $a = \mu(\mathbb{X})$.
  $$\begin{pmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{pmatrix}, \begin{pmatrix} y_1 \\ \dots \\ y_\ell \end{pmatrix} \xrightarrow{\mu} a.$$

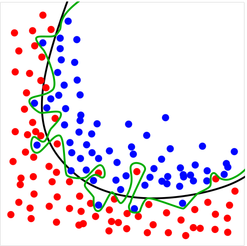
- Применение модели: получение ответов на новых данных.
  $$\begin{pmatrix} f_1(x'_1) & \dots & f_n(x'_1) \\ \dots & \dots & \dots \\ f_1(x'_\ell) & \dots & f_n(x'_\ell) \end{pmatrix} \xrightarrow{a} \begin{pmatrix} a(x'_1) \\ \dots \\ a(x'_\ell) \end{pmatrix}.$$

Функционалы качества. Обучение модели

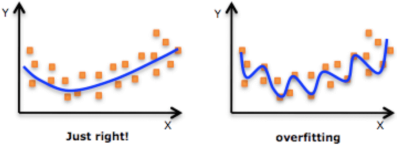
- Функция потерь $L(x, a)$ – величина ошибки алгоритма $a \in A$ на объекте x.

  Функция потерь для классификации:
  $$L(x, a) = [a(x) \neq y(x)].$$
  Функции потерь для задачи регрессии:
  $$L(x, a) = (a(x) - y(x))^2, \quad L(x, a) = |a(x) - y(x)|.$$
- Эмпирический риск:
  $$Q(a, \mathbb{X}) = \frac{1}{\ell} \cdot \sum_{i=1}^{\ell} L(x_i, a).$$
- Минимизация эмпирического риска:
  $$a^* = \min_{a \in A} Q(a, \mathbb{X}).$$

Проблема переобучения (overfitting)



(a) Источник: Википедия



(b) Источник: datarobot.com

- Причина переобучения: семейство алгоритмов слишком гибкое («лишние» степени свободы тратятся на запоминание шума в данным).
- Как обнаружить переобучение: разделить обучающую выборку на две части: train и test. Обучить модель на train, оценить качество на test.

Вероятностная постановка задачи классификации

- X – множество объектов,
  Y – множество ответов.
- Совместное распределение:
  $$p(x, y) = P(y)p(x|y).$$
  $P_y = P(y)$ – априорные вероятности классов,
  $p_y(x) = p(x|y)$ – функции правдоподобия классов.
- Цель. По известным плотностям распределения $p_y(x)$ и априорным вероятностям P_y всех классов Y построить алгоритм a^*, минимизирующий вероятность ошибочной классификации:
  $$a^* = \operatorname*{argmin}_a \mathbb{P}(a(x) \neq y).$$

Оптимальный байесовский классификатор

- Задача. Найти оптимальный алгоритм a^*:

$$a^* = \operatorname*{argmin}_a \mathbb{P}(a(x) \neq y) = \operatorname*{argmax}_a \mathbb{P}(a(x) = y).$$

- Решение (принцип максимума апостериорной вероятности):

$$a^*(x) = \operatorname*{argmax}_{y \in Y} P(y|x) = \operatorname*{argmax}_{y \in Y} \frac{P(y)P(x|y)}{P(x)} =$$

$$= \operatorname*{argmax}_{y \in Y} P(y)P(x|y).$$

- Доказательство:

$$\mathbb{P}(a(x) = y) = \mathbb{E}\,\mathbb{P}(a(x) = y|x) \leqslant \mathbb{E}\,\mathbb{P}(a^*(x) = y|x) = \mathbb{P}(a^*(x) = y).$$

Наивный байесовский классификатор

- Предположение. Условная независимость признаков при условии класса:

$$P(f_1(x), f_2(x), \dots, f_n(x)|y) = \prod_{k=1}^{n} P(f_k(x)|y).$$

- Оптимальный алгоритм:

$$a^*(x) = \operatorname*{argmax}_{y\in Y} P(y|x) = \operatorname*{argmax}_{y\in Y} P(y)P(x|y) =$$

$$= \operatorname*{argmax}_{y\in Y} P(y) \prod_{k=1}^{n} P(f_k(x)|y).$$

Как бороться с обнулением вероятностей?

Модель bag of words (мешок слов)

- Множество объектов – коллекция текстов.
- Каждый текст принадлежит одному из классов.
- Модель генерации текста: каждое слово появляется в тексте независимо от остальных. Порядок слов никак не учитывается.
- Вероятности появления слов зависят от класса.

Построение решающего правила:

- Оценить априорные вероятности классов (доли классов в выборке).
- Для каждого класса оценить вероятность появления слова.
- Воспользоваться принципом максимума апостериорной вероятности.

Гипотезы непрерывности и компактности

- Гипотеза компактности (классификация): похожие объекты лежат в одном классе.
- Гипотезка непрерывности (регрессия): близким объектам соответствуют близкие ответы.

Метрики

На множестве X задана метрика $\rho(x, x')$:

1. $\rho(x, x') = \rho(x', x)$.
2. $\rho(x, x') = 0$ тогда и только тогда, когда $x = x'$.
3. $\rho(x, x') \leqslant \rho(x, z) + \rho(z, x')$ для любого $z \in X$.

Примеры «расстояний». Расстояние Минковского

Расстояние Минковского ($x, x' \in \mathbb{R}^n$):

$$\rho(x, x') = \left(\sum_{k=1}^{n} |x_k - x'_k|^r \right)^{1/r}, \quad r \geqslant 1.$$

Примеры:

1. Евклидово расстояние ($r = 2$):

$$\rho(x, x') = \left(\sum_{k=1}^{n} |x_k - x'_k|^2 \right)^{1/2}.$$

2. Манхэттенское расстояние ($r = 1$):

$$\rho(x, x') = \sum_{k=1}^{n} |x_k - x'_k|.$$

3. Расстояние Чебышева ($r = +\infty$):

$$\rho(x, x') = \max_{1 \leqslant k \leqslant n} |x_k - x'_k|.$$

Что будет при $r \to 0$?

Примеры «расстояний». Расстояние Минковского

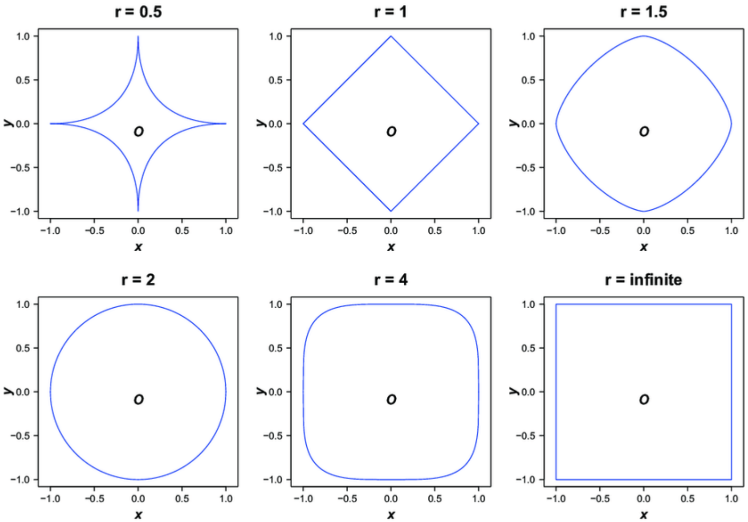


Рис. 4: Источник: researchgate.net

Еще примеры сходств и расстояний

- Косинусное сходство:

$$\mathrm{sim}(x, x') = \frac{x^T x'}{\|x\| \cdot \|x'\|}.$$

- Расстояние Жаккара между множествами A и B:

$$\rho(A, B) = 1 - \frac{A \cap B}{A \cup B}.$$

- Расстояние Левенштейна: минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую.

Обобщенный метрический классификатор

Обозначим через $x_u^{(i)}$ – i-й сосед объекта $u \in X$:

$$\rho(u, x_u^{(1)}) \leqslant \rho(u, x_u^{(2)}) \leqslant \ldots \leqslant \rho(u, x_u^{(\ell)}).$$

Ответ на i-м соседе:

$$y_u^{(i)} = y(x_u^{(i)}).$$

Обобщенный метрический классификатор:

$$a(u, \mathbb{X}) = \operatorname*{argmax}_{y \in Y} \sum_{i=1}^{\ell} [y_u^{(i)} = y] \cdot w(i, u),$$

где $w(i, u)$ – оценка степени важности i-го соседа для классификации объекта u.

Метод ближайших соседей

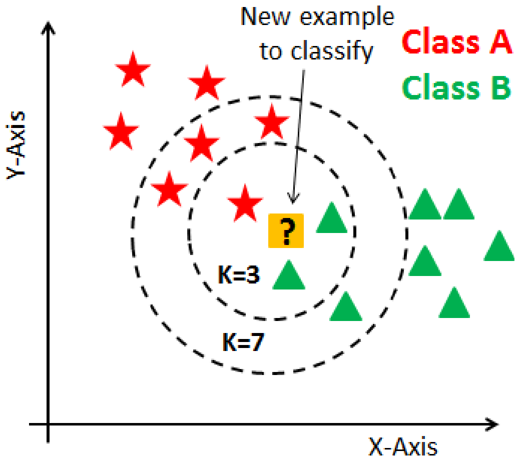


Рис. 5: Источник:kdnuggets.com

Метод ближайших соседей

Обобщенный метрический классификатор:

$$a(u, \mathbb{X}) = \operatorname*{argmax}_{y \in Y} \sum_{i=1}^{\ell} [y_u^{(i)} = y] \cdot w(i, u).$$

В зависимости от выбора весовой функции получаем различные метрические алгоритмы классификации.

- Метод одного ближайшего соседа:

$$w(i, u) = [i = 1], \quad a(u, \mathbb{X}) = y_u^{(1)}.$$

- Метод k ближайших соседей:

$$w(i, u) = [i \leqslant k], \quad a(u, \mathbb{X}) = \operatorname*{argmax}_{y \in Y} \sum_{i=1}^{k} [y_u^{(i)} = y].$$

Проблемы:

- Никак не учитываются расстояния до ближайших объектов.
- Максимум может достигаться на нескольких классах.

Как подбирать k? Как предобрабатывать данные? Что лучше: $k = 1$, $k = 2$? Как гибкость алгоритма зависит от k?

Метод k взвешенных ближайших соседей

- Весовая функция:

$$w(i, u) = [i \leqslant k] \cdot w_i,$$

где w_i – вес, зависящий только от номера i.

- Возможные подходы:
  1. Линейно убывающие веса:

$$w_i = \frac{k + 1 - i}{k}.$$

  2. Экспоненциально убывающие веса:

$$w_i = q^i, \quad 0 < q < 1.$$

Метод парзеновского окна

- Функция ядра $K(r)$ – неотрицательная невозрастающая функция на $[0, +\infty]$.
- Метод парзеновского окна фиксированной ширины:

$$w(i, u) = K\left(\frac{\rho(u, x_u^{(i)})}{h}\right),$$

$$a(u, \mathbb{X}, h, K) = \underset{y \in Y}{\operatorname{argmax}} \sum_{i=1}^{\ell} [y_i = y] \cdot K\left(\frac{\rho(u, x_i)}{h}\right),$$

где $h > 0$ – ширина окна.

- Метод парзеновского окна переменной ширины:

$$w(i, u) = K\left(\frac{\rho(u, x_u^{(i)})}{\rho(u, x_u^{(k+1)})}\right),$$

$$a(u, \mathbb{X}, k, K) = \underset{y \in Y}{\operatorname{argmax}} \sum_{i=1}^{\ell} [y_i = y] \cdot K\left(\frac{\rho(u, x_i)}{\rho(u, x_u^{(k+1)})}\right).$$

Метод потенциальных функций

- Ширина окна зависит не от классифицируемого объекта, а от объекта выборки.
- Метод потенциальных функций:

$$w(i,u) = \gamma(x_u^{(i)}) \cdot K\left(\frac{\rho(u, x_u^{(i)})}{h(x_u^{(i)})}\right),$$

$$a(u, \mathbb{X}) = \underset{y \in Y}{\operatorname{argmax}} \sum_{i=1}^{\ell} \gamma_i \cdot K\left(\frac{\rho(u, x_i)}{h_i}\right),$$

где $\gamma_i \geqslant 0$ – веса объектов, $h_i > 0$ – ширина окна i-го объекта.

- Пример функции ядра:

$$K(r) = \frac{1}{r+a}, \quad a \geqslant 0.$$

Метод k ближайших соседей в задаче регрессии

- Общий вид (взвешенное среднее):

$$a(u, \mathbb{X}) = \frac{\sum\limits_{i=1}^{\ell} w(i,u) \cdot y_u^{(i)}}{\sum\limits_{i=1}^{\ell} w(i,u)}.$$

- Классический алгоритм k ближайших соседей:

$$w(i,u) = [i \leqslant k],$$

$$a(u, \mathbb{X}) = \frac{1}{k} \cdot \sum_{i=1}^{k} y_u^{(i)}.$$

- Формула Надарая–Ватсона:

$$w(i,u) = K\left(\frac{\rho(u, x_u^{(i)})}{h(x_u^{(i)})}\right),$$

$$a(u, \mathbb{X}) = \frac{\sum\limits_{i=1}^{\ell} K\left(\frac{\rho(u, x_u^{(i)})}{h(x_u^{(i)})}\right) \cdot y_u^{(i)}}{\sum\limits_{i=1}^{\ell} K\left(\frac{\rho(u, x_u^{(i)})}{h(x_u^{(i)})}\right)}.$$

Проклятие размерности

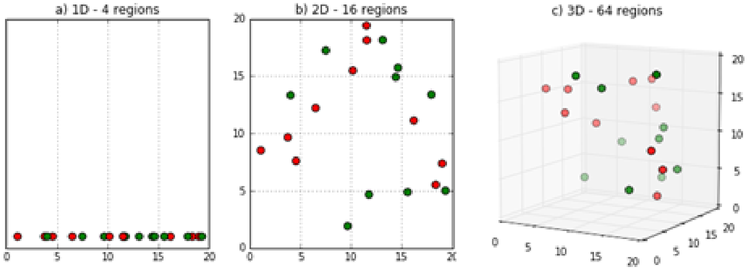


Рис. 6: Источник: deepai.org

- С ростом количества признаков данные становятся все более разреженными.
- Количество требуемых данных экспоненциально возрастает с ростом количества признаков.

Какова вероятность попасть в куб $[0, 0.99]^d$?

Поиск ближайших соседей

Библиотеки для быстрого приближенного поиска ближайших соседей:

- Annoy (Spotify)
- Faiss (Facebook)

Резюме лекции

- Вероятностная постановка задачи классификации.
  - Оптимальный байесовский классификатор.
  - Наивный байесовский классификатор.
- Метрические методы классификации и регрессии.
  - Гипотезы непрерывности и компактности.
  - Обобщенный метрический классификатор.
  - Метод k ближайших соседей.
  - Метод k взвешенных ближайших соседей.
  - Метод парзеновского окна.
  - Метод потенциальных функций.
  - Метод k ближайших соседей в задаче регрессии.
  - Проклятие размерности.