Метрические методы классификации

Машинное обучение, 20!7

Малютин Е. А.

Содержание

Сегодня в программе:

- Основные понятия и определения
- Гипотеза компактности
- Метод ближайших соседей и его обобщения
- Окно Парзена
- Отступы и выбор эталонов
- Почему kNN так хорошо работает
- Оценка качества классификации

Определения 1

Пусть $\exists \ \{x_1, x_2 ... x_l\} \ \in X$ – *множество объектов* (1)

И $\exists \ \{y_i\}_{i=1}^{l} \in Y$ – *множество допустимых ответов* (2)

Пары (x_i, y_i) – называются прецедентами, (3)

совокупность пар $X^l = (x_i, y_i)_{i=1}^{l}$ – обучающая выборка, (4)

а так же существует зависимость (алгоритм): $y^* : X \to Y$ (5)

Определения 2

Типы задач

- $Y = \{1, ..., M\}$ – задача классификации
- $Y = 0, 1$ – задача бинарной классификации
- $Y = \mathbb{R}$ – задача регрессии
- $y^* : y^*(x, t)$ – задача прогнозирования
- Y нет – задача обучения без учителя

Ещё определения

- . Моделью алгоритмов называется параметрическое семейство отображений $A = \{g(x, \theta) | \theta \in \Theta\}$, где $g : X \times \theta \to Y$ – некоторая фиксированная функция, θ – множество допустимых значений параметра θ, называемое пространством параметров или пространством поиска (search space)

И ещё определения

- Функция потерь (loss function) — это неотрицательная функция $L(a, x)$, характеризующая величину ошибки алгоритма a на объекте x. Если $L(a, x) = 0$, то ответ $a(x)$ называется корректным.
- А величину $Q = \frac{1}{l}\sum_{i=1}^{l} \mathcal{L}(a, x_i)$ – функционал качества.

Функционалы потерь

- $\mathcal{L}(a, x) = \{0 \mid 1\}$ – частота ошибок
- $\mathcal{L}(a, x) = [a(x) \neq y(x)]$ – индикатор ошибки, обычно применяется в задачах классификации
- $\mathcal{L}(a, x) =\mid a(x) - y(x) \mid$ — отклонение от правильного ответа; функционал Q называется средней ошибкой алгоритма a на выборке X^l
- $\mathcal{L}(a, x) = (a(x) - y(x))^2$ – квадратичная функция потерь; функционал Q называется средней квадратичной ошибкой алгоритма a на выборке X^l; обычно применяется в задачах регрессии

Метрические методы

Гипотеза компактности

Гипотеза компактности – в задачах классификации предположение о том, что схожие объекты гораздо чаще лежат в одном классе, чем в разных;

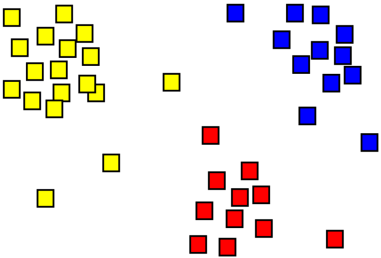

Рис.: Выполнение гипотезы компактности

Метрические методы

Задача машинного обучения

Пусть на множестве объектов X задана функция расстояния $\rho: \ X \times X \ \rightarrow \ [0, \infty)$. Существует целевая зависимость $y^*: \ X \ \rightarrow \ Y$, значения которой известны только на объектах обучающей выборки $X^l = (x_i, y_i)_{i=1}^l, y_i = y^*(x_i)$

Обобщённый метрический алгоритм классификации

$\alpha(\mu, X^l) = arg \max\limits_{y \in Y} \mathcal{J}_y(\mu, X^l); \quad \mathcal{J}_y(\mu, X^l) = \sum_{i=1}^l [y_\mu^l w(i, \mu)];$

Соседи

Алгоритм ближайшего соседа

Относим объект к классу, которому принадлежит ближайший обучающий объект.

+ Простаяа
- Неустойчив к погрешностям, выбросам
- Отсутствуют параметры, которые можно было бы настроить
- Низкое качество классификации

Соседи 2

Метод k ближайших соседей

Алгоритм k ближайших соседей – относит объект к классу, элементов которого больше среди k ближайших соседей.

Проблема:

Однозначные ответы;

Решение:

Ввод строго убывающей последовательности вещественных весов, задающих вклад соседа в классификацию.

Что ещё не так?

- Приходится хранить обучающую выборку целиком.
- В наивной реализации приходится тратить $O(l)$ времени
- Ну и ещё проблемы с границами классов

Соседи 3

Метод Парзеновского окна

Рассмотрим след. вариант:

$\alpha(\mu; X^l, h) = arg \max\limits_{y \in Y} \sum_{i=1}^{l} [y_\mu^i = y] K(\frac{\rho(\mu, x_\mu^i)}{h})$ – способ задать весовую функцию $w(i, u)$ как функция от ядра K, невозрастающую на $[0, \infty]$ Фиксация ширины окна h – не подходит всегда. В этих случаях применяется окно переменной ширины:

$\alpha(\mu; X^l, h) = arg \max\limits_{y \in Y} \sum_{i=1}^{l} [y_\mu^i = y] K(\frac{\rho(\mu, x_\mu^i)}{\rho(\mu, x_\mu^{(i+1)})})$

Об объектах

Введём понятие **отступа**: $M(x_i) = \mathcal{J}_{y_i}(x_i) - \max\limits_{y \in Y \setminus y_i} \mathcal{J}_{y_i}(x_i)$ (y_i – класс, J – «близость» к элементам относительно класса y_i)

Типы объектов

- *Эталонные объекты* – имеют большой положительный отступ, наиболее типичны.
- *Неинформативные объекты* – также имеют положительный отступ.
- *Пограничные объекты* – неустойчивая классификация.
- *Ошибочные объекты* – имеют отрицательные отступы и классифицируются неверно.
- *Шумовые* – это небольшое число объектов с большими отрицательными отступами.

Картинки

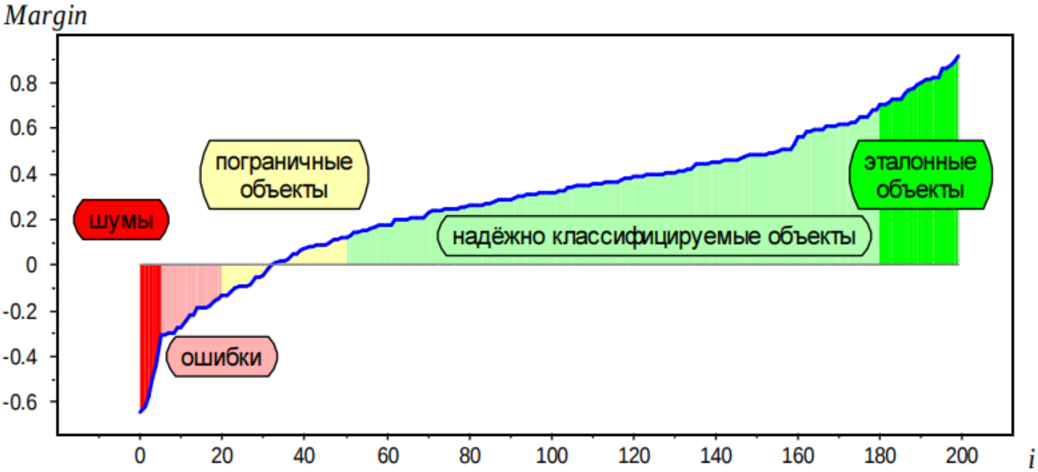

Рис.: Упорядоченные по возрастанию отступов M_i объекты выборки, $i = 1, ..., 200$. Условное деление объектов на пять типов.

Функционалы ошибки

В прекрасной России будущего $X \times Y$ является вероятностным пространством с плотностью распределения $p(x, y) = P(y)p(x|y)$.

Виды штрафов (некоторые):

- $L(y, y') = (y' - y)^2$ – регрессия
- $L(y, y') = \begin{cases} 0, y = y' \\ 1, y \neq y' \end{cases}$ – классификация
- $R(f) = \int L(f(x), y) dP(x, y)$ – ожидаемая ошибка предсказания (средний риск):

Принцип минимизации среднего риска

Голос со стороны учебника мат. стата:

1. По имеющейся выборке восстанавливаем функцию распределения $P(x, y)$.
2. Полученная функция $\hat{P}(x, y)$ подставляется под интеграл и решается задача вариационного исчисления.

Как восстанавливать?

- Выборочная функция распределения
- Параметрические методы
- Непараметрические методы

Выборки, валидации, оценки

$R(f) \approx \hat{R}(f) = \frac{1}{l}\sum L(f(x), y)$ – эмпирический риск.

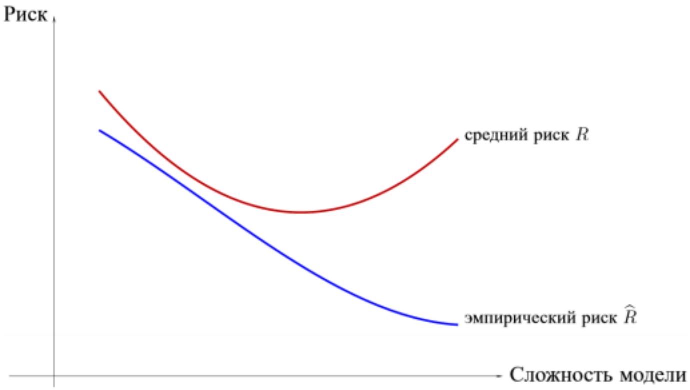

Рис.: Связь между средним и эмпирическим в прекрасной России будущего

Выборки, валидации, оценки

Выборки

- Обучающая (Train) – для построения моделей
- Проверочная (Validation) – для оценки среднего риска для каждой из построенных моделей и выбора из них оптимальной.
- Тестовая (Test) – для оценки ошибки предсказания выбранной модели.

Выборки, валидации, оценки

Ошибки

- Истинно-положительное решение (TP)
- Истинно-отрицательное решение (TN)
- Ошибка 1-го рода – ложно-положительное решение (FP)
- Ошибка 2-го рода – ложно-отрицательное решение (FN)

«Волки-волки»(гипотеза – "волка нет"):
когда крестьяне прибежали в первый раз – они совершили ошибку 1-го рода;
а вот когда парня съели – ошибку второго.

Выборки, валидации, оценки

Классификация

- $Accuracy = \frac{TP+TN}{N}$
- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- $F_{score} = (\beta^2 + 1)\frac{P \times R}{(\beta^2 * P)+R}$
- $P_{mac} = \frac{1}{K} \sum \frac{TP_i}{TP_i+FP_i}$ $\quad R_{mac} = \frac{1}{K} \sum \frac{TP_i}{TP_i+FN_i}$
- $P_{mic} = \frac{1}{K} \frac{\sum TP_i}{\sum(TP_i+FP_i)}$ $\quad R_{mic} = \frac{1}{K} \sum \frac{\sum TP_i}{\sum(TP_i+FN_i)}$

Выборки, валидации, оценки

Перекрёстный:

- Выборка разбивается на 2 части
- Модель обучается на одной, «тестируются» – на другой
- Выборки меняются местами
- Усредняем метрики
- PROFIT!1!!1!

K-fold валидация:

- Выборка разбивается на K части
- В цикле каждая из K считается тестовой, остальные – train
- Собираем K метрик
- Усредняем метрики

И ещё валидация:

Если вам мало:

- PROFIT!1!!1!
- \+ Точнее (насколько – расскажу потом)
- Можно вообще $K = l$, получаем самый точный способ оценки - LOO (leave one out)
- — но и самый дорогой

Практика:

SEMION:

- Считать
- Распарсить
- Нарисовать на matplotlib
- Собраь KNN на scikit-learn
- Проанализировать результат
- Покачать гиперпараметры
- Показать результат