

Методы классификации. Часть 1. Оценка качества.

Кантонистова Елена

elena.kantonistova@yandex.ru

30 ноября 2017

- 1 Постановка задачи классификации
- 2 Метод k ближайших соседей (KNN)
- 3 Наивный байесовский классификатор
- 4 Логистическая регрессия
- 5 Метрики качества классификации

Постановка задачи классификации

Пусть X - пространство объектов (матрица признаков), Y - вектор ответов (в случае бинарной классификации значения Y могут быть $\{+1, -1\}$). Необходимо построить функцию $a(x) : X \rightarrow Y$, которая для любого объекта из X предсказывает ответ.

Примеры задач классификации

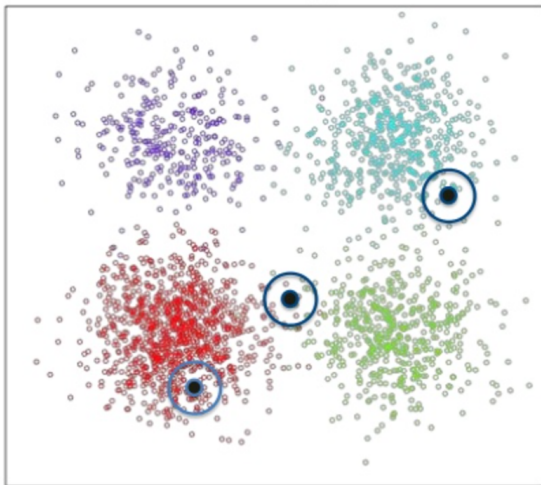
- Предсказание пола для неизвестного пользователя
- Предсказание категории письма: спам/не спам, важное/не важное
- Вероятность ухода сотрудника/клиента
- Вероятность невозврата кредита
- Определение языка для неизвестного текста
- Определение темы новостного сообщения
- Определение объекта на фотографии: человек, дом, автомобиль,
...
- Определение эмоциональной окраски твита
- Определение состояния человека по ЭКГ

Качество алгоритма измеряет функционал качества или же функционал ошибки. Самый очевидный функционал качества при классификации:

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

$Q(a, X)$ - доля правильных ответов алгоритма.

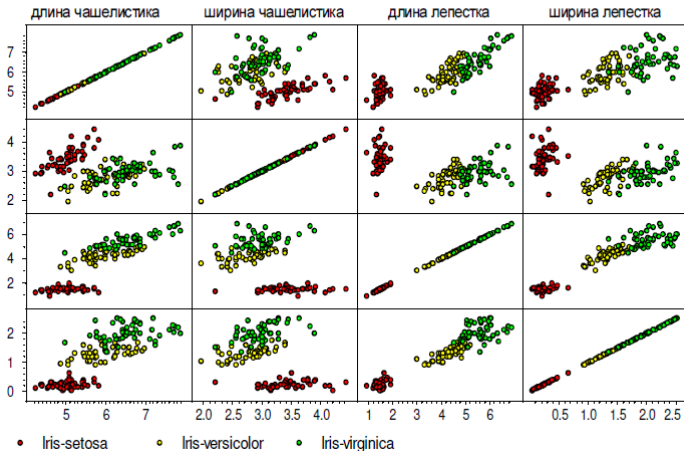
Метод k ближайших соседей



Гипотеза компактности. Близкие объекты, как правило, лежат в одном классе.

Пример

$n = 4$ признака, $|Y| = 3$ класса, длина выборки $\ell = 150$.



Математическая формулировка KNN

Отранжируем объекты выборки по расстоянию до данного объекта x :

$$\text{dist}(x, x^{(1)}) \leq \text{dist}(x, x^{(2)}) \leq \text{dist}(x, x^{(3)}) \dots$$

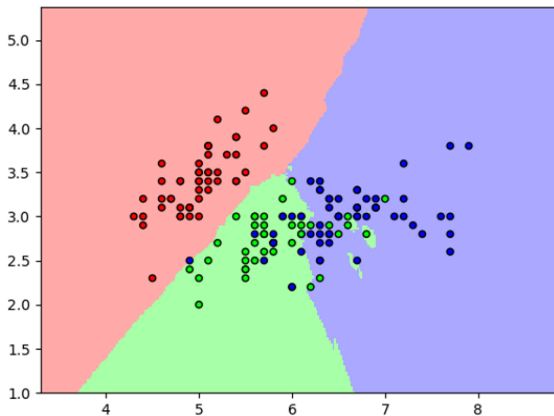
$x^{(i)}$ - объект выборки, $y^{(i)}$ - его класс.

Класс объекта x определяется по формуле:

$$a(x; X) = \underset{y \in Y}{\operatorname{argmax}} \sum_{i=1}^k [y^{(i)} = y],$$

то есть объект x относим к тому классу, представителей которого вокруг него наибольшее число.

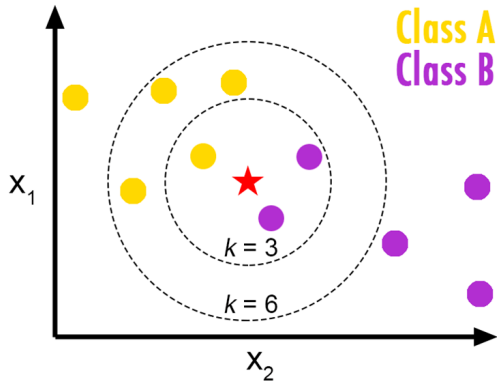
Пример



Подбор k - количества соседей

Число соседей k , на которое мы ориентируемся - гиперпараметр алгоритма. Как его выбирать?

Выбор числа k

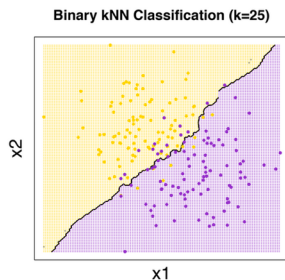
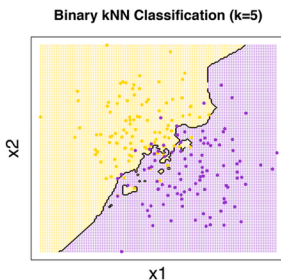
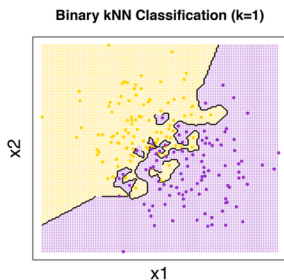


Параметр k подбираем по кросс-валидации.

- Разбиваем выборку на тренировочную, валидационную и тестовую часть
- Обучаем алгоритм для различных k на тренировочной части
- Выбираем такой k , чтобы качество на валидационной части было наилучшим

Геометрическая интерпретация k

Чем больше k - тем проще устроена граница между классами.



Плюсы:

- Простота метода
- Интерпретируемость решений

Минусы:

- Неустойчивость к погрешностям данных (к шуму, выбросам)
- Малое число параметров, следовательно, слабые модели
- Как правило, низкое качество классификации

- Наивный байесовский алгоритм – это алгоритм классификации, основанный на теореме Байеса с допущением о независимости признаков.
- Другими словами, НБА предполагает, что наличие какого-либо признака в классе не связано с наличием какого-либо другого признака.

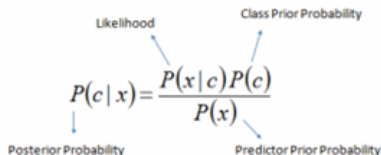
Пример: фрукт может считаться яблоком, если:

- он красный
- круглый
- его диаметр составляет порядка 8 сантиметров

Даже если эти признаки зависят друг от друга или от других признаков, в любом случае они вносят независимый вклад в вероятность того, что этот фрукт является яблоком.

Теорема Байеса

Теорема Байеса позволяет рассчитать апостериорную вероятность $P(c|x)$ на основе $P(c)$, $P(x)$ и $P(x|c)$.



The diagram shows the formula for Bayes' Theorem: $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$. Arrows point from labels to the corresponding parts of the formula: 'Likelihood' points to $P(x|c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c|x)$, and 'Predictor Prior Probability' points to $P(x)$.

$$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$$

- $P(c|x)$ – апостериорная вероятность данного классас(т.е. данного значения целевой переменной) при данном значении признаках
- $P(c)$ – априорная вероятность данного класса
- $P(x|c)$ – правдоподобие, т.е. вероятность данного значения признака при данном классе
- $P(x)$ – априорная вероятность данного значения признака.

Рассмотрим обучающий набор данных, содержащий один признак «Погодные условия» (weather) и целевую переменную «Игра» (play), которая обозначает возможность проведения матча.

На основе погодных условий мы должны определить, состоится ли матч.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Преобразуем набор данных в частотную таблицу (frequency table).

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Таблица частот

Создадим таблицу правдоподобия (likelihood table), рассчитав соответствующие вероятности.

Likelihood table				
Weather	No	Yes		
Overcast		4	$=4/14$	0.29
Rainy	3	2	$=5/14$	0.36
Sunny	2	3	$=5/14$	0.36
All	5	9		
	$=5/14$	$=9/14$		
	0.36	0.64		

Мы можем решить эту задачу с помощью описанного выше подхода.

$$P(Yes|Sunny) = P(Sunny|Yes) * P(Yes)/P(Sunny)$$

Мы можем решить эту задачу с помощью описанного выше подхода.

$$P(\text{Yes}|\text{Sunny}) = P(\text{Sunny}|\text{Yes}) * P(\text{Yes})/P(\text{Sunny})$$

Здесь мы имеем следующие значения:

- $P(\text{Sunny}|\text{Yes}) = 3/9 = 0,33$
- $P(\text{Sunny}) = 5/14 = 0,36$
- $P(\text{Yes}) = 9/14 = 0,64$

Применение формулы Байеса

Мы можем решить эту задачу с помощью описанного выше подхода.

$$P(Yes|Sunny) = P(Sunny|Yes) * P(Yes)/P(Sunny)$$

Здесь мы имеем следующие значения:

- $P(Sunny|Yes) = 3/9 = 0,33$
- $P(Sunny) = 5/14 = 0,36$
- $P(Yes) = 9/14 = 0,64$

Теперь рассчитаем $P(Yes|Sunny)$:

$$P(Yes|Sunny) = 0,33 * 0,64 / 0,36 = 0,60$$

Значит, при солнечной погоде более вероятно, что матч состоится.

Аналогичным образом с помощью наивного байесовского алгоритма можно прогнозировать несколько различных классов на основе множества признаков. Этот алгоритм в основном используется в области классификации текстов и при решении задач многоклассовой классификации.

Линейный классификтор проводит разделяющую плоскость в пространстве признаков.

Пусть x_1, x_2, \dots - признаки. Задача линейного классификатора - оптимальным образом подобрать коэффициенты (веса) при признаках: w_1, w_2, \dots Линейный классификатор:

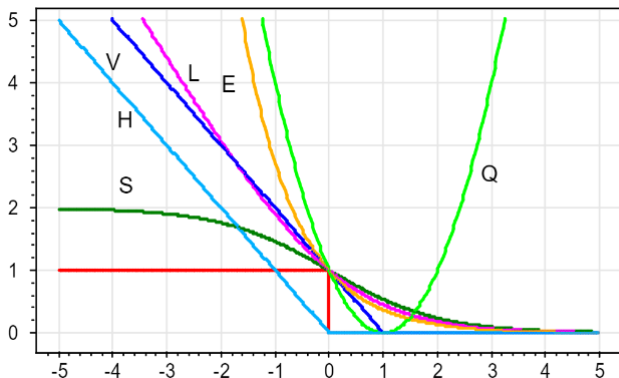
$$a(x) = \text{sign}(w_1x_1 + w_2x_2 + \dots)$$

Необходимо подобрать веса w_1, w_2, \dots оптимальным образом, то есть чтобы минимизировать функцию потерь (долю неправильных ответов):

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

Она не является гладкой!

Гладкие функции потерь



Логистическая функция потерь

Функция потерь, используемая алгоритмом логистической регрессии:

$$L(M) = \log(1 + e^{-M})$$

Логистическая регрессия корректно оценивает вероятности принадлежности к классам.

Пусть в каждой точке x задана вероятность $P(y = +1|x)$, то есть вероятность того, что объект x принадлежит классу $+1$. То есть в выборке могут быть объекты с одинаковыми признаками, но разными классами.

Пример. При посещении одного и того же сайта пользователь может кликнуть по одному и тому же баннеру, а может не кликнуть.

Минимизация функции потерь

Мы минимизируем логистическую функцию потерь, которая на одном объекте выглядит так:

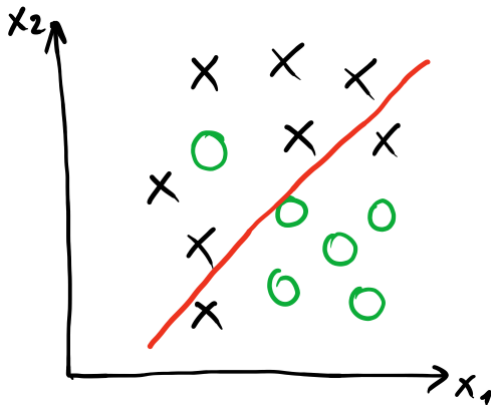
$$L(M) = L(y, z) = \log(1 + e^{-yz}).$$

Оказывается, что минимум функционала потерь на данном объекте достигается при таких значениях весов z , что

$$\arg \min_{b \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^n L(y_i, b) \approx p(y = +1 | x).$$

Линейный классификатор

Важно помнить, что логистическая регрессия - это классификатор!
Причем линейный.



Accuracy - доля правильных ответов классификации.

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

Истинный ответ	0	0	1	1	1
Ответ модели	1	0	1	1	0

Accuracy = ?

Accuracy - доля правильных ответов классификации.

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

Истинный ответ	0	0	1	1	1
Ответ модели	1	0	1	1	0

$$Accuracy = 3/5 = 0.6$$

Пример: кредитный скоринг

Всего было 200 человек. Первая модель выдала 100 кредитов, вторая - 50.

Модель 1:

- 80 кредитов вернули
- 20 кредитов не вернули

Модель 2:

- 48 кредитов вернули
- 2 кредита не вернули

Какая модель лучше?

Что хуже?

- Выдать кредит «плохому» клиенту
- Не выдать кредит «хорошему» клиенту

Доля верных ответов не учитывает цены ошибок

Матрица ошибок

	$y = 1$ Клиенты, которые вернут кредит	$y = -1$ Клиенты, которые не вернут кредит
$a(x) = 1$ Модель «сработала» Решение о выдаче кредита	True Positive (TP) Клиенты, которые получили кредит и вернули его	False Positive (FP) Клиенты, которые получили кредит и не вернули его
$a(x) = -1$ Модель выдала «пропуск» Решение об отказе в кредите	False Negative (FN) Клиенты, которые не получили кредит, но при этом могли бы его вернуть	True Negative (TN) Клиенты, которые не получили кредит и не вернули бы его

Матрица ошибок

	$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть		$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть
$a(x) = 1$ Получили кредит	80	20	$a(x) = 1$ Получили кредит	48	2
$a(x) = -1$ Не получили кредит	20	80	$a(x) = -1$ Не получили кредит	52	98

Точность (precision) -

показывает, насколько можно ли доверять классификатору при $a(x) = +1$.

Для кредитного скоринга вычисляется как доля клиентов, получивших кредит и вернувших его, среди всех клиентов, получивших кредит

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

Precision

	$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть		$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть
$a(x) = 1$ Получили кредит	80	20	$a(x) = 1$ Получили кредит	48	2
$a(x) = -1$ Не получили кредит	20	80	$a(x) = -1$ Не получили кредит	52	98

Для первой модели точность 0.8, для второй - 0.96.

Полнота показывает, как много положительных объектов находит классификатор.

Для кредитного скоринга вычисляется как доля клиентов, получивших кредит, среди всех клиентов, которые могли бы его вернуть

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

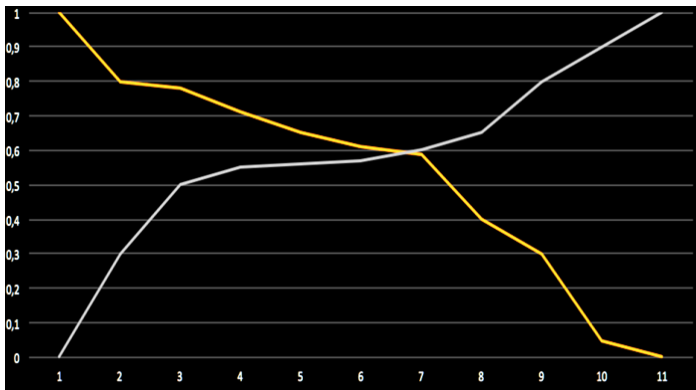
Recall

	$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть		$y = 1$ Могут вернуть	$y = -1$ Не могут вернуть
$a(x) = 1$ Получили кредит	80	20	$a(x) = 1$ Получили кредит	48	2
$a(x) = -1$ Не получили кредит	20	80	$a(x) = -1$ Не получили кредит	52	98

Для первой модели полнота 0.8, для второй - 0.48.

Точность и полнота

Точность и полнота измеряют два различных аспекта качества. При росте одной из метрик другая, как правило, уменьшается



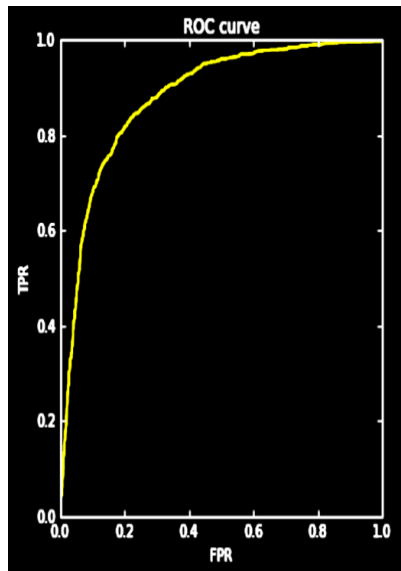
Частая ситуация:

классификатор оценивает вероятность принадлежности к
положительному классу

Примеры: кредитный скоринг, медицинская диагностика и т.д

Необходимо выбирать порог.

Как оценить качество классификатора без конкретного порога?



- Левая точка: $(0, 0)$, Правая точка: $(1, 1)$
- Для идеального классификатора проходит через $(0, 1)$
- AUC-ROC — площадь под ROC-кривой
- AUC-ROC около 0.5 — плохой классификатор
- AUC-ROC = 1 — идеальный классификатор
- $Gini = 2 * AUC - 1$