Машинное обучение

Лекция 4. Задача классификации, логистическая регрессия

Автор: Рустам Азимов

Санкт-Петербург, 2023г.

Задача предсказания

- Предсказание значения количественного признака Y называется **задачей регрессии**
- Предсказание значения номинального (категориального) признака Y называется **задачей классификации**
- Например, предсказание признака Возраст — это задача регрессии, а предсказание признака Пол — задача классификации

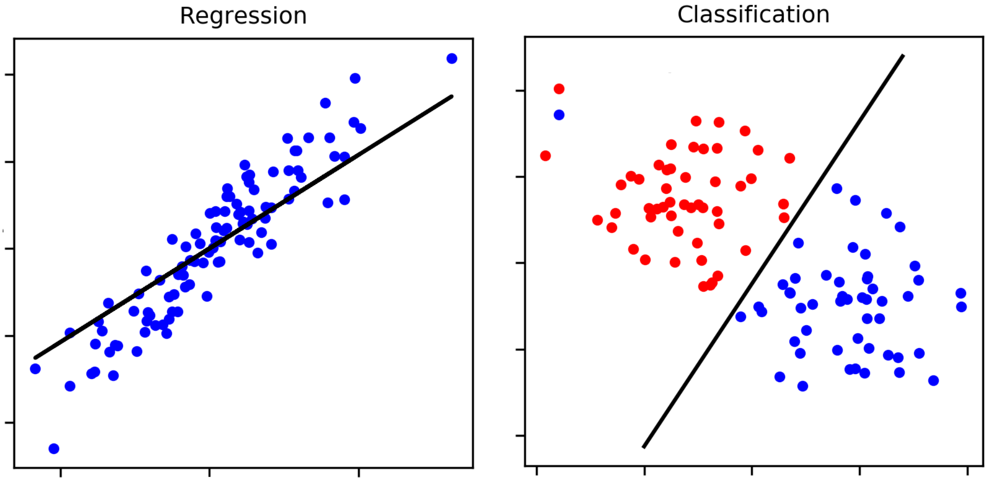
Задача бинарной классификации

- X — признаки, вещественные числа
- Целевой признак $Y \in \{-1, +1\}$
- Линейная модель для классификации:

$$\alpha(x) = sign(\sum_{i=0}^{m} w_i x_i) = sign\langle w, x\rangle$$

- Разделяет пространство на две части гиперплоскостью
- Величина скалярного произведения описывает расстояние до гиперплоскости, а его знак — по какую сторону данный объект

Regression vs Classification



Accuracy

- Если мы хотим угадать все классы в обучающей выборке, то мы легко можем переобучиться
- Будем считать **accuracy** — долю правильных ответов модели на обучающей выборке с векторами признаков $\vec{x}_i$ и соответсвующими им ответами y_i:

$$\frac{1}{n}\sum_{i=1}^{n}[\alpha(\vec{x}_i) = y_i]$$

- Для функции ошибок модели α, которую будем минимизировать подходит доля неправильных ответов:

$$Q(\alpha, X) = \frac{1}{n}\sum_{i=1}^{n}[\alpha(\vec{x}_i) \neq y_i] = \frac{1}{n}\sum_{i=1}^{n}[sign\langle w, \vec{x}_i\rangle \neq y_i]$$

- Так как эта функция дискретна относительно весов w, то градиентный спуск для решения не подходит
- Может быть много глобальных минимумов

Отступ

- Знак характеристики **отступ (margin)** говорит о корректности ответа классификатора:
$$M_i = y_i \langle w, \vec{x}_i \rangle$$
- Положительный знак — ответ правильный, отрицательный — неправильный
- А само значение отступа характеризует уверенность нашего классификатора в своём ответе
- Чем больше значение — тем больше уверенность

Отступ

- Тогда функцию ошибки можно переписать в виде:

$$Q(\alpha, X) = \frac{1}{n}\sum_{i=1}^{n}[\alpha(\vec{x}_i) \neq y_i] = \frac{1}{n}\sum_{i=1}^{n}[sign\langle w, \vec{x}_i\rangle \neq y_i] = \frac{1}{n}\sum_{i=1}^{n}[M_i < 0]$$

- При значениях M_i близким к нулю, данный объект находится близко к гиперплоскости нашего классификатора, из-за чего у модели низкая уверенность в ответе

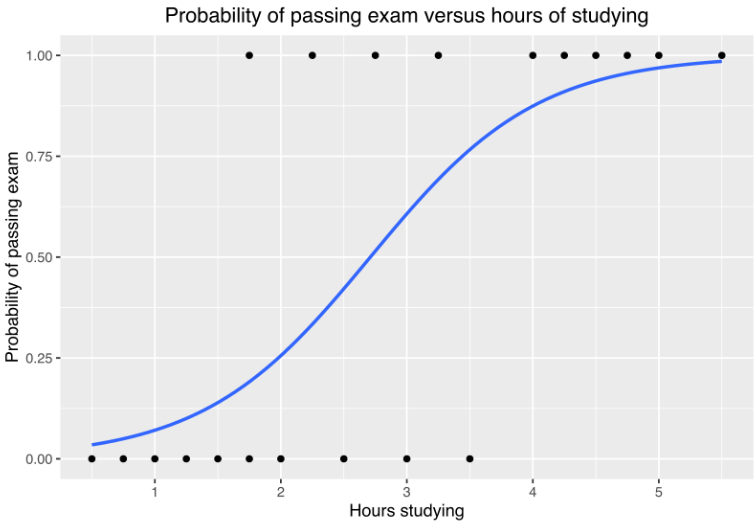
Функция потерь бинарного классификатора

- Вместо того чтобы работать с кусочно-линейной функцией $L(M) = [M < 0]$ можно заменить её на гладкую верхнюю оценку этой функции $\bar{L}(M) \geq L(M)$
- Например, может быть использована сигмоида: **логистическая функция потерь**

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

- Увидим, что такая функция подходит для обучения линейного классификатора

Logistic Function



Оценка качества бинарного классификатора

- После обучения модели необходимо оценить её качество
- Рассмотрим более общий вид нашего классификатора:

$$\alpha(x) = [b(x) > t]$$

- В случае линейной модели $b(x) = \langle w, x \rangle$ и $t = 0$

Оценка качества бинарного классификатора

- Как уже говорилось, очевидной функцией является доля правильных ответов модели:
$$accuracy(\alpha, x) = \frac{1}{n}\sum_{i=1}^{n}[\alpha(\vec{x}_i) = y_i]$$
- Чем плоха оценка *accuracy*?

Оценка качества бинарного классификатора

- Как уже говорилось, очевидной функцией является доля правильных ответов модели:

$$accuracy(\alpha, x) = \frac{1}{n}\sum_{i=1}^{n}[\alpha(\vec{x}_i) = y_i]$$

- Чем плоха оценка *accuracy*?
- Допустим мы взяли порог t меньше минимального значения прогноза $b(x)$ на обучающей выборке или больше максимального
- Тогда доля правильных ответов будет равна доле положительных и отрицательных ответов соответственно
- Если в выборке 990 отрицательных и 10 положительных объектов, то $accuracy(\alpha, x) = 0,99$, хотя предсказатель глупый

Оценка качества бинарного классификатора

- Поэтому полезно также анализировать соотношение классов в обучающей выборке
- Вычисляют **базовую долю** — долю правильных ответов модели, которая всегда предсказывает наиболее мощный класс

Матрица ошибок

- Чтобы оценка была более информативной, рассмотрим следующие критерии

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

Precision/recall

- Тогда доля правильных ответов выражается:

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

- Но более информативными являются **точность (precision)** и **полнота (recall)**

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

Precision/recall

- Точность показывает, какая доля объектов, которые по мнению классификатора являются положительными, действительно положительны
- Полнота показывает, какая доля из положительных объектов была угадана классификатором
- Можно регулировать точность и полноту, изменяя порог t в классификаторе $\alpha(x) = [b(x) > t]$
- Если выбрать большое t, то классификатор будет относить к положительному классу небольшое число объектов и, следовательно, точность будет высокой, а полнота — низкой
- При уменьшении t точность будет падать, а полнота — увеличиваться
- Конкретное значение порога выбирается согласно пожеланиям заказчика

Пример

- Задача — предсказать реакцию клиента оператора сотовой связи на звонок с предложением подключить новую услугу
- Положительные объекты (клиенты) с $y = 1$ это те, кто примут предложение после рекламного звонка
- Отрицательные с $y = -1$ это те, кто не примут
- Пытаемся предсказать с помощью классификатора кто примет предложение, а кто нет и будем звонить тем, для кого $\alpha(x) = 1$
- Важнее точность или полнота?

Пример

- Задача — предсказать реакцию клиента оператора сотовой связи на звонок с предложением подключить новую услугу
- Положительные объекты (клиенты) с $y = 1$ это те, кто примут предложение после рекламного звонка
- Отрицательные с $y = -1$ это те, кто не примут
- Пытаемся предсказать с помощью классификатора кто примет предложение, а кто нет и будем звонить тем, для кого $\alpha(x) = 1$
- Важнее точность или полнота?
- При высокой точности обученного классификатора практически каждый звонок будет результативным для оператора сотовой связи
- А при высокой полноте — звонки покроют практически всех целевых клиентов

F-мера

- Точность и полнота не зависят от соотношения размеров классов в обучающей выборке, но с приходится оперировать двумя критериями
- Вместо этого можно использовать один критерий, например, **F-меру** — гармоническую среднюю точности и полноты:

$$F = \frac{2 * precision * recall}{precision + recall}$$

- Гармоническое среднее близко к нулю, если хотя бы один из аргументов близок к нулю
- F-мера является сглаженной версией минимума из точности и полноты

Оценка качества семейства моделей

- Мы рассмотрели как оценивать классификатор $\alpha(x) = [b(x) > t]$ при известной функции $b(x)$ и выбранном пороге t
- Но зачастую, порог будет выбираться позже в зависимости от требований к точности и полноте
- Поэтому мы хотим оценивать качество сразу семейства моделей

$$\{\alpha(x) = [b(x) > t \mid t \in \mathbb{R}]\}$$

AUC-ROC

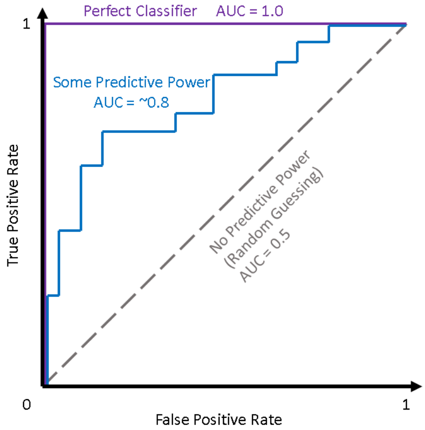
- Широко используется такая интегральная метрика качества семейства, как площадь под **ROC-кривой (Area Under ROC Curve, AUC-ROC)**
- Чтобы изобразить эту кривую нам нужны FPR и TPR
- Доля неверно принятых объектов **(False Positive Rate, FPR)**

$$FPR = \frac{FP}{FP + TN}$$

- И доля верно принятых объектов **(True Positive Rate, TPR)**

$$TPR = \frac{TP}{TP + FN}$$

AUC-ROC


Perfect Classifier AUC = 1.0
Some Predictive Power
AUC = ~0.8
No Predictive Power
(Random Guessing)
AUC = 0.5
True Positive Rate
False Positive Rate
1
0
1

AUC-ROC

- Каждый возможный выбор порога t соответствует точке в этом пространстве
- Всего различных порогов $n + 1$, отсортируем объекты $\vec{x_i}$ по возрастанию функции $b(x)$
- Максимальный порог $t_{max} = max_i b(\vec{x_i})$ даст классификатор с $TPR = 0$ и $FPR = 0$
- Минимальный порог $t_{min} = min_i b(\vec{x_i}) - \varepsilon$ даст классификатор с $TPR = 1$ и $FPR = 1$
- **ROC-кривая** — это кривая с концами в точках $(0, 0)$ и $(1, 1)$, которая последовательно соединяет точки, соответствующие порогам $b(\vec{x_1}) - \varepsilon, b(\vec{x_1}), b(\vec{x_2}), \dots, b(\vec{x_n})$

Значение AUC-ROC

- Площадь под этой кривой называется **AUC-ROC** и принимает значения от 0 до 1 и чем больше, тем качественней классификатор
- При близости площади к 0.5 классификатор ранжирует объекты случайным образом
- Если площадь меньше 0.5, то предсказывать наоборот выгоднее

Логистическая регрессия

- Метод обучения, который получается при использовании логистической функции потерь, называется **логистической регрессией**

$$\bar{L}(y, \langle w, x \rangle) = log(1 + e^{-y\langle w, x \rangle})$$

- Он позволяет корректно оценивает вероятность принадлежности объекта к каждому из классов, например, $p(y = +1 \mid x)$
- Далее покажем как можно вывести эту функцию ошибок с помощью метода максимального правдоподобия

Максимальное правдоподобие

- Мы хотим построить $b(x)$ предсказывающую вероятность принадлежности объекта к положительному классу, то есть пусть $b(x)$ имеет область значений $[0,1]$
- Наша задача выбрать такую функцию $b(x)$, что правдоподобие выборки (т.е. вероятность получить такую выборку с точки зрения фукнции $b(x)$) будет максимальным

$$P(\alpha, X) = \prod_{i=1}^{n} b(\vec{x}_i)^{[y_i=+1]}(1 - b(\vec{x}_i))^{[y_i=-1]} \to \max$$

- А точнее удобней перейти к минимизации и к логарифму (чтобы оптимизировать сумму, а не произведение)

$$-\sum_{i=1}^{n} \left([y_i = +1] log(b(\vec{x}_i)) + [y_i = -1] log(1 - b(\vec{x}_i))\right) \to \min$$

Log-loss

- Получившаяся функция ошибки или потерь называется **логарифмической (log-loss)**
- Её можно использовать для обучения
- Оптимальный ответ равен вероятности положительного класса

Сигмоидная функция

- Мы требовали, чтобы $b(x)$ имела область значений $[0, 1]$
- Но для линейного классификатора у нас $b(x) = \langle w, x \rangle$
- Применяем любую монотонно неубывающую функцию с областью значений $[0, 1]$
- Мы будем использовать сигмоидную (логистическую) функцию:

$$\sigma(\langle w, x \rangle) = \frac{1}{1 + e^{-\langle w, x \rangle}}$$

- Таким образом,

$$p(y = +1 \mid x) = \frac{1}{1 + e^{-\langle w, x \rangle}}$$

Log-odds

- Выразим скалярное произведение

$$\langle w, x \rangle = log \frac{p(y = +1 \mid x)}{p(y = -1 \mid x)}$$

- Оно равно логарифму отношения вероятностей классов (**log-odds**)

Вывод функции потерь

- Подставим сигмоидную функцию в функцию потерь для правдоподобия

$$-\sum_{i=1}^{n}\left([y_i = +1]log\frac{1}{1+e^{-\langle w,\vec{x}_i\rangle}} + [y_i = -1]log\frac{e^{-\langle w,\vec{x}_i\rangle}}{1+e^{-\langle w,\vec{x}_i\rangle}}\right)$$

$$= -\sum_{i=1}^{n}\left([y_i = +1]log\frac{1}{1+e^{-\langle w,\vec{x}_i\rangle}} + [y_i = -1]log\frac{1}{1+e^{\langle w,\vec{x}_i\rangle}}\right)$$

$$= \sum_{i=1}^{n} log(1+e^{-y_i\langle w,\vec{x}_i\rangle})$$

Логистические потери

- Полученная функция в точности представляет собой логистические потери
- Линейная модель классификации, настроенная путём минимизации данного функционала, называется **логистической регрессией**
- Она оптимизирует правдоподобие выборки и даёт корректные оценки вероятности принадлежности к положительному классу

Многоклассовая классификация

- X — признаки, вещественные числа
- Целевой признак $Y \in \{1, \dots, k\}$, k — количество классов
- Есть много способов свести к серии бинарных задач оценки вероятности принадлежности к положительному классу

One-versus-all

- **Один против всех (one-versus-all)**: обучим k линейных классификаторов $b_1(x), \dots, b_k(x)$, выдающих оценки принадлежности классам $1, \dots, k$ соответственно
- Классификатор с номером j будем обучать по выборке $(x_i, 2 * [y_i = j] - 1)_{i=1}^n$
- То есть мы учим классификатор отличать j-ый класс от всех остальных
- Итоговый классификатор будет выдавать класс, соответствующий самому уверенному из бинарных алгоритмов, то есть с наибольшим $b_j(x)$

Дополнительные источники

- machinelearning.ru
- scikit-learn.org
- kaggle