Лабораторная работа № 3. Функции ошибок в машинном обучении

Цель работы

Получение знаний и критериев применимости основных используемых в современном машинном обучении функций ошибок (функций потерь).

Краткие теоретические сведения

Функция потерь — функционал, оценивающий величину расхождения между истинным значением оцениваемого параметра и модельной оценкой этого параметра.

Задачи регрессии

Выбирая функцию потерь для задач регрессии, следует решить, какое именно свойство условного распределения мы хотим восстановить. Наиболее частые варианты:

$$L(y, f) = (y - f)^2$$

– Gaussian loss (*L2*), самый часто используемый и простой вариант, если нет никакой дополнительной информации или требований к устойчивости модели.

$$L(y, f) = |y - f|$$

– Laplacian loss (*L1*); в некоторых задачах эта функция потерь предпочтительнее, так как она не так сильно штрафует большие отклонения, нежели квадратичная функция.

$$L(y, f) = \begin{cases} (1 - \alpha)|y - f| & \text{при } y - f \leq 0 \\ \alpha|y - f| & \text{при } y - f > 0 \end{cases}$$

– Quantile loss (Lq), функция асимметрична и больше штрафует наблюдения, оказывающиеся по нужную сторону квантили.

Графики перечисленных функций приведены на рисунке 12.

Задачи классификации

При классификации из-за принципиально другой природы распределения целевой переменной оптимизируются не сами метки классов, а их *log*-

правдоподобие. Наиболее известные варианты таких классификационных функций потерь изображены на рисунке 13.

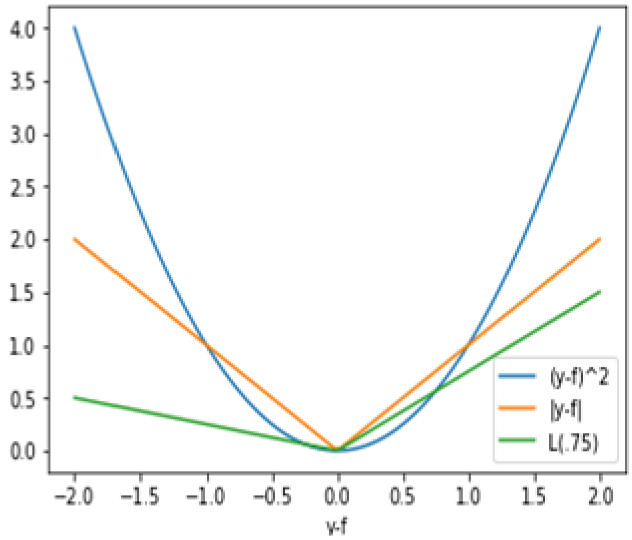

Рисунок 12 – Функции потерь для задач регрессии

Математические выражения для расчёта этих функций следующие:

$$L(y, f) = \ln(1 + \exp(-yf))$$

– Logistic loss (Bernoulli loss). Штрафуются даже корректно предсказанные метки классов, но, оптимизируя эту функцию потерь, можно улучшать классификатор, даже если все наблюдения предсказаны верно. Это самая стандартная и часто используемая функция потерь в бинарной классификации.

$$L(y, f) = \exp\,(-yf)$$

– Adaboost loss; имеет более жесткий экспоненциальный штраф на ошибки классификации и используется реже.

Для использования в задачах классификации применима также функция кросс-энтропии, которая определяет меру расхождения между двумя вероятностными распределениями. Если кросс-энтропия велика, разница между двумя распределениями велика, а если кросс-энтропия мала, распределения похожи друг на друга:

$$H(P, Q) = -\sum P(x) \cdot \log_2 Q(x),$$

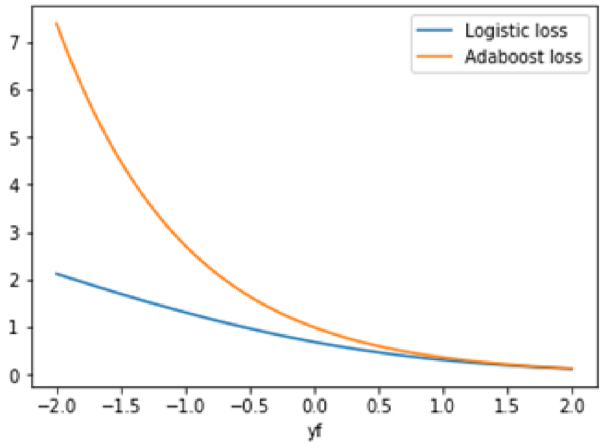

Рисунок 13 – Функции Logistic loss и Adaboost loss

где P – распределение истинных ответов, а Q – распределение вероятностей прогнозов модели.

В случае бинарной классификации формула для расчёта кросс-энтропии имеет следующий вид:

$$L = -\mathrm{y} \cdot \log_2 p + (1 - y) \cdot \log_2(1 - p),$$

где y – двоичный индикатор (0 или 1) того, является ли метка класса правильной классификацией для текущего наблюдения, p – прогнозируемая вероятность класса, определённая моделью классификации.

При бинарной классификации каждая предсказанная вероятность сравнивается с фактическим значением класса (0 или 1) и вычисляется оценка, которая штрафует вероятность пропорционально величине отклонения от ожидаемого значения. Конфигурация выходного уровня модели представляет собой один узел с сигмоидальной функцией активации:

$$f(p_i) = \frac{1}{1 + \exp{(-p_i)}}$$

Чтобы классифицировать объект как принадлежащий одному из нескольких классов, задача формулируется как предсказание вероятности того, что пример принадлежит каждому классу. В случае, когда классов много (*M>2*) берется сумма значений логарифмических функций потерь для каждого прогноза наблюдаемых классов – «categorical cross-entropy»:

$$CE = -\sum_{c=1}^{M} y_{o,c} \cdot \log_2 p_{o,c}.$$

Для расчёта взвешенного вектора вероятностей отнесения объекта к конкретным классам используется функция активации «softmax» — обобщение логистической функции для многомерного случая.

$$f(p_i) = \frac{\exp{(p_i)}}{\sum_{j=1}^{M} \exp{(p_j)}}.$$

Когда каждый из объектов должен классифицироваться однозначно, что чаще всего и требуется, можно отбросить слагаемые, которые являются нулевыми из-за значений целевых индикаторов y. Тогда:

$$CE = -\log_2 \frac{\exp{(p_i)}}{\sum_{j=1}^{M} \exp{(p_j)}},$$

где p_i – результат оценки принадлежности к соответствующему классу.

Ход работы

1. Скачать данные:

2. Реализовать модель логистической регрессии со следующими функциями потерь:
   а) Logistic loss
   б) Adaboost loss
   в) binary crossentropy
3. Визуализировать кривые обучения модели бинарной классфикации в виде динамики изменения каждой из функций ошибок п.2 на тренировочной и тестовой выборках.

4. Сравнить качество классификации по метрике accuracy в каждом из трёх модификаций алгоритма.

Дополнительные вопросы и задания

1. Какую функцию потерь нужно применять в задаче регрессии, если вы хотите, чтобы модель больше штрафовала за выбраосы в данных?

2. Как функция кросс-энтропии связана с дивергенцией Кульбака-Лейблера?

Литература

1. Основные функции потерь и примеры их реализации на Python: https://www.analyticsvidhya.com/blog/2019/08/detailed-guide-7-loss-functions-machine-learning-python-code/

2. Алгоритм градиентного бустинга с разбором функций потерь для регрессии и классификации: https://habr.com/ru/company/ods/blog/327250/#3-funkcii-poter