Домашнее задание №5: «Индейцы пима, диабет и линейный классификатор»

Дедлайн 1 (20 баллов): 13 апреля, 23:59
Дедлайн 2 (10 баллов): 20 апреля, 23:59

Домашнее задание нужно написать на Python и сдать в виде одного файла. Правило именования файла: `name_surname_5.[py | ipnb]`. Например, если вас зовут Иван Петров, то имя файла должно быть: `ivan_petrov_5.py` или `ivan_petrov_5.ipnb`.

Рис. 1: Источник: `http://gilariver.org`

Индейцы племени пима проживают в центральной и южной части штата Аризона. По неизвестным на данный момент причинам индейцы пима имеют критический риск заболевания сахарным диабетом (2 типа). В этом задании предлагается применить линейный классификатор для диагностики и изучения сахарного диабета (2 типа) у представителей племени пима.

По ссылке[1] находятся медицинские данные представителей племени пима. Последняя колонка каждой строки — индикатор наличия сахарного диабета (2 типа). Значения остальных колонок указаны в заголовке файла.

[1] `https://gist.github.com/ktisha/c21e73a1bd1700294ef790c56c8aec1f`

1 Реализуйте функцию `read_data`, которая принимает путь к файлу с данными и возвращает пару из двух массивов NumPy[2].

- Первый элемент пары — матрица признаков `X`, в первой колонке которой находится константный признак -1, а в остальных признаки из файла с данными.
- Второй элемент пары — вектор `y`, в котором -1 означает наличие диабета, а 1 — его отсутствие.

Возможно, вам будет полезна функция `genfromtext`, читающая матрицу NumPy из CSV файла.

2 Реализуйте несколько вариантов функции потерь. Функция потерь должна принимать на вход **вектор** отступов `M` и возвращать пару из вектора значений функции потерь и вектора её производных. Например, если бы мы решили использовать степенную функцию потерь:

```
def power_loss(M, n=5):
    return M ** (-n), -n * (M ** (-n - 1))
```

Необходимо реализовать как минимум логарифмическую (`log_loss`) и сигмоидную (`sigmoid_loss`) функции потерь.

3 Реализуйте метод градиентного спуска для обучения линейного классификатора в виде класса `GradientDescent`. Структура класса приведена ниже:

```
class GradientDescent:
    def __init__(self, *, alpha, threshold=1e-2, loss=sigmoid_loss):
        if alpha <= 0:
            raise ValueError("alpha should be positive")
        if threshold <= 0:
            raise ValueError("threshold should be positive")
        self.alpha = alpha
        self.threshold = threshold
        self.loss = loss

    def fit(self, X, y):
        errors = []
        # ...
        return errors

    def predict(self, X):
        # ...
```

Метод `fit` должен:

- случайно инициализировать веса,
- оценить веса линейного классификатора с использованием функции потерь,
- записать полученные веса в атрибут `self.weights`
- и вернуть список значений функционала качества $Q(\boldsymbol{w})$ на каждой итерации градиентного спуска.

[2] Хорошее введение в API массивов NumPy можно найти на сайте `http://scipy-lectures.github.io/intro/numpy`

Результатом метода `predict` является вектор предсказаний из -1 и 1, полученный с использованием весов, оцененных в методе `fit`.

В качестве критерия остановки следует использовать отсечку `threshold` на расстояние между векторами весов на текущей и предыдущей итерациях. Расстояние можно выбрать любое, например, Евклидово или ℓ^1.

Обратите внимание, что на данных индейцев пима алгоритм должен работать **не более** 30 секунд.

4 Реализуйте метод стохастического градиентного спуска для обучения линейного классификатора. Структура класса аналогична:

```
class SGD:
    def __init__(self, *, alpha, loss=log_loss, k=1, n_iter=100):
        if alpha <= 0:
            raise ValueError("alpha should be positive")
        if k <= 0 or not isinstance(k, int):
            raise ValueError("k should be a positive integer")
        if n_iter <= 0 or not isinstance(n_iter, int):
            raise ValueError("n_iter should be a positive integer")

        self.k = k
        self.n_iter = n_iter
        self.alpha = alpha
        self.loss = loss

    def fit(self, X, y):
        errors = []
        # ...
        return errors

    def predict(self, X):
        # ...
```

Как и в случае с `GradientDescent` метод `fit` должен возвращать значения Q на каждой итерации. Значение $\eta \in [0, 1]$, используемое для вычисления оценки Q, можно выбрать любое, например, `1 / len(X)`. Так как величина Q не стабильна, использовать её для определения сходимости не следует. Вместо этого предлагается использовать "стратегию оптимиста": сделать ровно `n_iter` итераций и надеяться, что за это время стохастический градиентный спуск сойдётся.

Параметр `k` определяет размер случайной подвыборки из `X`, используемой для вычисления градиента. Такой вариант стохастического градиентного спуска называется *mini-batch*. При `k = 1` он вырождается в алгоритм, описанный на лекции.

Обратите внимание, что на данных индейцев пима 1000 итераций алгоритма должна занимать **не более** 30 секунд.

5 Воспользуйтесь функциями `test_train_split` и `print_precision_recall` из домашнего задания 2 для оценки качества работы реализованных алгоритмов на данных индейцев пима.

6 Исследуйте чувствительность алгоритмов к выбору параметров.

- Постройте график функционала качества $Q(\boldsymbol{w})$ алгоритма `GradientDescent` в зависимости от значения $\alpha \in \{10^{-6}, 10^{-4}, 10^{-2}, 1\}$.
- Постройте аналогичный график для оценки значения функционала качества Q алгоритма `SGD` в зависимости от α и $k \in \{1, 10, 50\}$.

Графики необходимо построить для всех реализованных вами функций потерь и приложить к письму.

7 Ответьте на вопросы с использованием графиков.

- Почему функционал качества убывает в процессе работы градиентного спуска? Почему это не всегда так для стохастического градиентного спуска?
- Как зависит скорость сходимости (количество итераций) алгоритмов от α?
- Как параметр k влияет на поведение Q в алгоритме стохастического градиентного спуска?
- Какие значения параметров каждого алгоритма вам кажутся оптимальными для задачи диагностики диабета? Почему?
- Как ведёт себя вектор весов $\boldsymbol{w}$ при повторных запусках алгоритмов? Объясните причины наблюдаемого поведения.
- Какой алгоритм уместнее использовать для данных индейцев пима? Почему?

8 Приведите наиболее симпатичный вам вектор весов $\boldsymbol{w}$ и дайте интерпретацию его значениями применительно к задаче диагностики диабета или объясните, почему это невозможно.