AI home test

August 31, 2020

1 Problem 1

Given N data points, distributed in 2-D plane as follow:

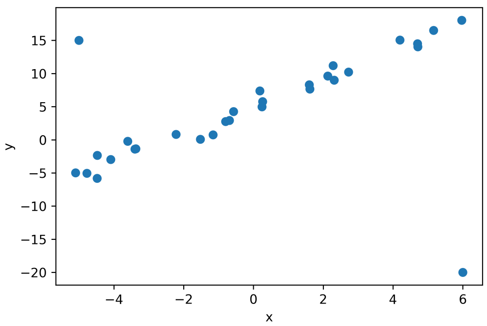

Figure 1: The input data

We would like to find a linear function that fits the given data.

$$\hat{y}_i = a \cdot x_i + b \tag{1}$$

You are required to implement gradient descent algorithm to solve the following optimization problem:

$$\text{minimize} \quad \sum_{i=1}^{N} \Phi(y_i - b - a \cdot x_i) \tag{2}$$

where

$$\Phi(r) = \begin{cases} \frac{1}{2}r^2, & \text{for } |r| \leq 1 \\ |r| - \frac{1}{2}, & \text{otherwise} \end{cases} \tag{3}$$

The data of 30 points are stored in `numpy` format, and can be loaded by:

```
import numpy as np
npzfile = np.load('data_hometest_1.npz')
x,y = npzfile['arr_0'], npzfile['arr_1']
```

You are expected to use only `numpy` Python package. If your implementation works well, the fitted line will look like Figure 2.

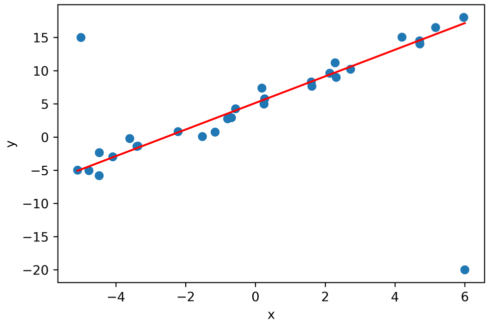

Figure 2: The input data

2 Problem 2

Product reviews are valuable for customers to make purchase decision. Very often helpful reviews are buried under huge amount of less helpful ones. Companies would like to exploit available data including helpfulness feedback of customers and to predict how helpful a review should be. Helpfulness is defined as the fraction of votes for helpfulness over total votes.

You are given Amazon review data [1] including reviews and product metadata of category **Office Products**, both are stored in JSON format. Review data contain essential information of reviews, for instance review content and ratings, helpfulness votes (in format $[a, b]$ where a is the number of votes for helpfulness over total number of votes b), and timestamps. Below is an example of review data.

[1] http://jmcauley.ucsd.edu/data/amazon/links.html

```
    "reviewText":"This is not coated in plastic as the
    ↪  description indicates.  It is very thin cardboard.  It is
    ↪  not a durable product and therefore I don\'t believe very
    ↪  useful for use around children.  I also thought there
    ↪  would be some way to attach the numbers, days, etc. and
    ↪  there is not.  I guess you would need to laminate this
    ↪  and then attach something to the backs to make them stick
    ↪  to a board (which again is not included).  It is just a
    ↪  bunch of punch outs, including the calendar portion which
    ↪  is in two pieces.",
    "reviewerName": "ANON",
    "helpful": [5, 5],
    "summary": "very disappointed",
    "unixReviewTime": 1389744000,
    "reviewerID": "A1BUVOGGFTGMBN",
    "reviewTime": "01 15, 2014",
    "overall": 2.0,
    "asin": "0439394058"
}
```

Product metadata, as a supplemental information, contain product descriptions, price, brand info, and co-purchasing links. They are mapped to review data through asin key. Below is an example of product metadata.

```
{
    "related": {
        "also_viewed": ["1594416249", "B004XMNTX6"],
        "also_bought": ["1594416249", "1600220991"]
    },
    "price": 9.99,
    "description": "Decorate for the Spring season. Includes 1
    ↪  Extra Large Easter Bunny (35&#x201D; tall), 4 Large
    ↪  Easter Characters, 1 &#x201C;Happy Easter&#x201D; Sign
    ↪  (48&#x201D; long), 8 Butterflies, and 36 Student Easter
    ↪  Eggs. 50 pieces.",
    "categories": [
        [
            "Office Products",
            "Office & School Supplies",
            "Education & Crafts",
            "Arts & Crafts Supplies",
            "Classroom Decorations"
        ]
    ],
    "salesRank": null,
    "asin": "0439492335",
    "imUrl":
    ↪  "http:\\/\\/ecx.images-amazon.com\\/images\\/I\\/61KmH%2BHcDSL._SX300_.jpg",
    "title": "Scholastic TF3078 Happy Easter! Bulletin Board",
    "brand": "Scholastic"
}
```

The data can be loaded from reviews_Office_Products.json.gz and meta_Office_Products.json.gz. You are expected to represent any solution you can come up with.