Homework Assignment 2

SDS 385 Statistical Models for Big Data

Please upload the HW on canvas by 10pm Nov 8th. Please type up your homework using latex. We will not accept handwritten homeworks[1].

1. Consider a design matrix $\boldsymbol{X}$ such that $\boldsymbol{X}$ has orthonormal columns, i.e. $\boldsymbol{X^T X} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the $p \times p$ identity matrix. Consider the following regularization:

$$\min_{\beta} \frac{1}{2}(\boldsymbol{X\beta - y})^T(\boldsymbol{X\beta - y}) + \lambda||\boldsymbol{\beta}||_m^p \tag{1}$$

   (a) Derive the solution to equation (1) for $m = 2, p = 2$ (ridge regression).

   (b) Derive the solution to equation (1) for $m = 1, p = 1$ (lasso).

   (c) When $m = 0, p = 1$, the penalty involves $||\boldsymbol{\beta}||_0 = \sum_{i=1}^p \mathbb{1}(\beta_i \neq 0)$.

      i. Is this a convex optimization problem? Why or why not?

      ii. Show that the solution to equation (1) with $m = 0$ is given by $\tilde{\boldsymbol{\beta}}$, where

$$\tilde{\beta}_i = \begin{cases} \boldsymbol{v_i^T y}, & \text{if } |\boldsymbol{v_i^T y}| > \sqrt{2\lambda} \\ 0 & \text{if } |\boldsymbol{v_i^T y}| \leq \sqrt{2\lambda} \end{cases}$$

      This is also called the hard thresholding estimator. $\boldsymbol{v_i}$ is the i^{th} column of $\boldsymbol{X}$.

2. Consider the following problem of fused Lasso, or total variation de-noising.

$$\min_x \frac{1}{2}\|x - z\|_2^2 + \lambda \sum_{i=1}^{n-1} |x_{i+1} - x_i|.$$

   Here, z is a noisy signal and λ is non-negative.

   (a) Write this problem as follows:

$$\min_x \frac{1}{2}\|x - z\|_2^2 + \lambda \|Dx\|_1.$$

   where D is a $n - 1 \times n$ matrix. What is D?

   (b) Write down the subgradient of the objective function.

   (c) Implement the subgradient descent algorithm. On the noisy.txt dataset, apply the algorithm and show the convergence plot against number of iterations. Play with different stepsizes and discuss how that affects the convergence.

[1] Two of these homeworks were adapted from A. Dimakis and C. Caramanis's class

(d) The problem can be written as

$$\min_x \frac{1}{2}\|x - z\|_2^2 + \lambda\|y\|_1$$
$$\text{s.t. } y = Dx \tag{2}$$

Show that the dual of the above problem is:

$$\min_u \frac{1}{2}u^T DD^T u - u^T Dz.$$
$$\text{s.t. } \|u\|_\infty \leq \lambda \tag{3}$$

(e) Now implement the proximal gradient algorithm for the dual problem. In order to do this, you may need to look at the proximal operator given below:

$$\text{prox}_t(x) = \arg\min_z \frac{\|x - z\|^2}{2t} + I_\lambda(z)$$

where $I_\lambda(z) = 0$ if $|z| \leq \lambda$ and ∞ otherwise. In this case, the proximal operator gets reduced to a projection operator. On the same datset, apply this algorithm and show the convergence plot against the number of iterations. How does this compare with the subgradient method you implemented before?

(f) Finally, for the dual proximal gradient method, show the effect of different λ values by plotting the denoised curve superimposed on the original noisy curve.

3. Download the dataset articles-1000.txt (1.6MB), which contains 1000 articles. Each row corresponds to an article, represented by an ID (e.g., t120) and its content (e.g., The Supreme Court in Johnnesberg on Friday...).

(a) Convert each article into a set of 2-shingles (bigrams). The goal is to map each article ID to a set[2] of IDs of shingles that article contains. You can do this by going over the data once as follows: for each article, first split[3] it into a list of words, remove the stopwords[4] , generate a list of 2-shingles, and then hash each shingle to a 32-bit integer (i.e., the shingle ID) using CRC32 hash. The resulting shingle IDs range from 0 to $2^{32} - 1$. What is the total number of unique shingles across all the books? What is the average number of shingles present per book?

Here is an example of a 2-shingle and its hashed value (i.e., the shingle ID):

```
import binascii
shingle = "machine learning"
shingleID = binascii.crc32(shingle) & 0xffffffff
```

(b) Generate MinHash signatures using 10 hash functions, where each is as follows: $h(x) = c_1 x + c_2 \bmod p$.

Set $p = 4294967311$, which is a prime number larger than $2^{32} - 1$. Uniformly sample 10 values of $c_1, c_2 \in \{1, 2, ..., p - 1\}$ and compute the corresponding MinHash signatures.

[2]You can use Python sets to do that. A set should contain unique elements.

[3]Use Python String split() method to do that– no need to remove the punctuations.

[4]you can remove the stop words specified by stopwords.words("english") from nltk.corpus.

For the 1st article, compare its signature vector with that of the rest of the articles and estimate their Jaccard similarity (i.e., compute the percentage of signatures that are equal). Find the book that has the largest (estimated) similarity with the 1st book then compute the actual Jaccard similarity between the two books (based on the computed shingle sets). Your result should be a triplet (bookID, estimatedJaccard, trueJaccard).

Hint: remember, you do not need to generate 10 permutations. Use the trick we learned in class, you can also find it in Section 3.3.5 of the "Mining of Massive Datasets" book linked from the class website.

(c) **Amplification** Use the LSH technique in Section 3.4 of the same book, construct $n = br$ MinHash signatures, where b is the number of bands, and r is the number of hash values in each band. Find all the articles that are "similar" to the 1st article (i.e., articles that agree in at least one band of signatures), and put it into a set called S (excluding the 1st article itself). Let $Sim(i, j)$ be the actual Jaccard similarity between articles i and j. Given a threshold t, define the percentage of false positives (fp) in set S as

$$\text{False positives} = \frac{|\{i \in S : Sim(i, 1) < t\}|}{|S|}.$$

Set t = 0.8, plot fp as a function of b (for $b = 1, 3, 5, 7, 9$ with $r = 2$). Similarly, plot fp as a function of r (for $r = 1, 3, 5, 7, 9$ with $b = 10$). For each (b, r) pair, plot the average value of fp over 10 realizations. Give a short comparison of the two.

(d) Find the 5 most similar article pairs without any approximation. How long did it take? Now find the 5 most similar article pairs using LSH (with b,r as hyperparameters). Compare your results for different (b, r) pairs.