Introduction to Machine Learning (67577)

Exercise 6
Convex analysis, SGD and regularization

June 12, 2019

Contents

Submission guidelines **2**

Theoretical Part **2**
- Warm Up . . . 2
- SVM . . . 2
- Logistic regression . . . 3
- Regularization . . . 3

Practical Part **4**
- SVM . . . 5
- Logistic regression . . . 6

Abstract

In the tirgul we learned about a special family of problems which we can solve efficiently- "convex problems". In this exercise, we are going to solve the "real world problem" of classifying handwritten digits by relexating it into a convex problem. In the theoretical part we will formulate relexation and prove that our relexated problem is convex, and in the practical part we will implement algorithm to solve this relexation efficiently.

Submission guidelines

- The assignment is to be submitted via Moodle only.
- Files should be zipped to **ex_6_First_Last.zip** (First is your first name, Last is your last name. In English, of course).
- The .zip file size should not exceed 10Mb.
- Each question is enumerated, use these numbers when answering your questions.

Theoretical Part

Warm Up

1. (5 points) **Prove:** Let $f_i : V \to \mathbb{R}$ for $i = 1, \ldots, m$ be functions and let $\gamma_1, \ldots, \gamma_m$ be positive scalars. Consider the function $g : V \to \mathbb{R}$ given by

$$g(u) = \sum_{i=1}^{m} \gamma_i f_i(u).$$

   If $f_1, \ldots, f_m$ are convex, then g is also convex.

2. (5 points) Give a counterexample for the following claim: Given two functions $f, g : \mathbb{R} \to \mathbb{R}$, define a new function $h : \mathbb{R} \to \mathbb{R}$ by $h = f \circ g$. If f and g are convex than h is convex as well.

3. (5 points) **Prove:** A function $f : C \to \mathbb{R}$ defined over a convex set C is convex iff its *epigraph* is a convex set, where epi(f) $= \{(u, t) : f(u) \leqslant t\}$.

4. (5 points) Let $f_i : V \to \mathbb{R}$, $i \in I$. Let $f : V \to \mathbb{R}$ given by $f(u) = \sup_{i \in I} f_i(u)$. If f_i are convex for every $i \in I$, then f is also convex. (Hint: what can you say about $epi(f)$?)

SVM

5. (5 points) Given $x \in \mathbb{R}^d$ and $y \in \{\pm 1\}$. Show that the hinge loss is convex in (w, b). Meaning, define

$$f(w, b) = l_{x,y}^{hinge}(w, b) = \max\left(0, 1 - y \cdot (\langle w, x \rangle + b)\right)$$

   show that f is convex in (w, b).

6. (5 points) Find a g such that $g \in \partial l_{x,y}^{hinge}(w, b)$.
   (Hint: you can find useful theorem about $\partial \underset{i \in [m]}{max} f_i(x)$ in the Tirgul)

7. (5 points) Given $f_1, \ldots, f_m : \mathbb{R}^d \to \mathbb{R}$ convex functions, and $\xi_k \in \partial f_k(x)$ for all k. Define $f : \mathbb{R}^d \to \mathbb{R}$ by $f(x) = \sum_{i=1}^{m} f_i(x)$. Show that $\sum_k \xi_k \in \partial \sum_k f_k(x)$.

8. (5 points) Given dataset $\{(x_i, y_i)\}_{i=1}^m \subset \mathbb{R}^d \times \{\pm 1\}$, $\lambda \geqslant 0$. Define the function $f : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}$ by:

$$f(w,b) = \frac{1}{m}\sum_{i=1}^{m} l_{x_i,y_i}^{hinge}(w,b) + \frac{\lambda}{2}\|w\|^2$$

Show that f is convex and find a member of the sub-gradient of f.

Logistic regression

Recall the logistic regression where we have $S = \{(x_i, y_i)\}_{i=1}^m$ and try to maximize the function:

$$L_S(w) = \prod_{i=1}^{m} l_{likelihood}(h_w, (x_i, y_i))$$

9. (4 points) Write $L_S(w)$ (You can take a look at tirgul 6) formally.

10. (4 points) Define:

$$\tilde{L}_S(w) = \sum_{i=1}^{m} y_i \left[-\log\left(\frac{1}{1+e^{-w^T x_i}}\right)\right] + (1-y_i)\left[-\log\left(1-\frac{1}{1+e^{-w^T x_i}}\right)\right]$$

Show that $\arg\max L_S$ is equivalent to $\arg\min_w \tilde{L}_S(w)$ (Hint: what can you say about $-\log(L_S)$?)

11. (Bonus: 5 points) Show that: $\nabla \tilde{L}_S(w) = \sum_{i=1}^m \left(\frac{1}{1+e^{-w^T x_i}} - y_i\right) x_i$.

12. (Bonus: 5 points) Show that:

$$H_{\tilde{L}_S(w)} = \sum_{i=1}^{m} x_i x_i^T \left(1 - \frac{1}{1+e^{-w^T x_i}}\right) \frac{1}{1+e^{-w^T x_i}}$$

($H_{\tilde{L}_S(w)}$ is the Hessian of $\tilde{L}_S(w)$).

13. (4 points) Use a second order condition for convexity in order to show that $\tilde{L}_S(w)$ is convex (Hint: for which values of x, w the term $\left(1 - \frac{1}{1+e^{-w^T x_i}}\right)\frac{1}{1+e^{-w^T x_i}}$ is positive? let $i \in [m]$, what can you say about the matrix $x_i x_i^T$? what can you say about the matrix $\sum_{i=1}^m x_i x_i^T$?).

Regularization

In this section we will show that although Ridge adds bias to the estimation of $\mathbf{w}$ it still decreases the MSE.

Let X be a **constant** d-by-m regression matrix and assume that XX^T is invertible. Let $\hat{\mathbf{w}}(\lambda) = \text{argmin}_{\mathbf{w}}\left(||\mathbf{y} - X^T\mathbf{w}||_2^2 + \lambda||\mathbf{w}||_2^2\right)$ be the ridge estimator and let $\hat{\mathbf{w}}(\lambda = 0) \equiv \hat{\mathbf{w}}$ be the least squares estimator.

- Assume the linear model is correct, namely $\mathbf{y} = X^T\mathbf{w} + \epsilon$ where $\epsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$.
- Recall that in this case: $\mathbb{E}[\hat{\mathbf{w}}] = \mathbf{w}$.

14. (4 points) Show that $\hat{\mathbf{w}}(\lambda) = A_\lambda \hat{\mathbf{w}}$ where
$$A_\lambda = (XX^T + \lambda I_d)^{-1}(XX^T)$$
15. (4 points) Conclude from the previous item that the Ridge estimator is biased for any $\lambda > 0$, i.e. show that $\lambda > 0 \Rightarrow \mathbb{E}[\hat{\mathbf{w}}(\lambda)] \neq \mathbf{w}$.
16. (4 points) Show that: $\text{Var}(\hat{\mathbf{w}}(\lambda)) = \sigma^2 A_\lambda (XX^T)^{-1} A_\lambda^T$.
Hint: Use the fact that for a non random matrix B and a random vector $\mathbf{z}$ we have $\text{Var}(B\mathbf{z}) = B\text{Var}(\mathbf{z})B^T$ and that $\text{Var}(\hat{\mathbf{w}}) = \sigma^2(XX^T)^{-1}$.
17. (Not trivial and quite technical bonus - 5 points) Derive explicit expressions for the (squared) bias and variance of $\hat{\mathbf{w}}(\lambda)$ as a function of λ, i.e. write a bias-variance decomposition for the mean square error of $\hat{\mathbf{w}}(\lambda)$. Show by differentiation that
$$\frac{d}{d\lambda}\text{MSE}(\lambda)|_{\lambda=0} = \frac{d}{d\lambda}\text{bias}^2(\lambda)|_{\lambda=0} + \frac{d}{d\lambda}\text{Var}(\lambda)|_{\lambda=0} < 0$$
18. (4 points) Conclude that, if the linear model is correct, a little Ridge regularization helps to reduce the MSE.

Practical Part

In this part we will solve classification problem on a handwritten digits dataset (the "Mnist" dataset- The dataset contains a 28 by 28 images of handwritten digits). Since the computer represent images by arrays of pixels (integer in the range [0,255]), we will describe each image as a vector. While there is a lot of important information in the 2d form of the image, we will see that in this specific case we can do well in the 1d form as well. So now the problem is reduced to finding an hypothesis from all hypotheses that classify vectors in $\{0...255\}^{mn}$ to two different classes. This hypotheses class is to big for as, so we take only "small" class: Half-spaces. When training the SVM model we are looking for separating hyperplane between the train samples from the two classes.

- Download the file load_data.py from the course website.
- Your results should contain your code, and all the figures you are asked to plot (in the PDF of the answers to the theoretical part). Additionally, please include a script ex6_runme.py that runs your code, generates the figures, and prints the parameters and errors you are asked to report.

Assumptions:

- When we write GD, we actually mean sub gradient descent.

- The GD (gradient descent) and SGD (stochastic gradient descent) algorithms will use the loss function defined in the theoretical part. While in SGD we define m as the batch size, in GD it will defined as the training set size (Meaning, the batch is the whole training set).
- At each iteration the SGD algorithm draws a batch of samples from the training set without repetitions (repetitions could occur between different batchs). In order to implement it examine the function- np.random.choice.
- We would like to test our hypothesis using the 0-1 loss, but in the training process we are going to use the hinge loss (or the likelihood) as a surrogate loss (**Train on hinge loss, test on 0-1 loss**).

Coding assignments:

SVM

19. (12 points, 4 for each section) Create a file: "SVM.py" that will contain the functions: gd, sgd, and test_error.

    (a) The declaration of the GD function should be: "*gd(data, label, iters, eta, w)*", where
    - *data*- a $n \times d$ numpy array, where n is the amount of samples, and d is the dimension of each sample
    - *labels*- a $n \times 1$ numpy array with the labels of each sample
    - *iters* - an integer that will define the amount of iterations.
    - *eta* - a positive number that will define the learning rate.
    - w- a $(d+1) \times 1$ numpy array, where d is the dimension of each sample- the parameters of the classifier (the last element is the bias).

    The function will output a $d \times 1$ numpy array, contain the output of the GD descent algorithm over 'iters' iterations. Use the gradient that we found in the SVM section of the theoretical part.

    (b) The declaration of the sgd function should be: "*sgd(data, label, iters, eta, w, batch)*", where in addition to the definitions from the previous bullet, we add:
    - *batch*- The amount of samples that the algorithm would draw and use at each iteration.

    The output of the function is the same as in gd.

    (c) The declaration of the *test_error* function should be: "*test_error(w, test_data, test_labels)*", where
    - w- a $(d+1) \times 1$ numpy array, where $h_w(x) = sign(\langle w[:-1], x\rangle + w[-1])$.
    - *test_data* - a $n \times d$ numpy array with n samples
    - *test_labels*- a $n \times 1$ numpy array with the labels of the samples.

    The function will output a scalar - with the respective 0-1 loss for the hypothesis.

20. (20 points, 4 for each section) After writing all this code, lets play with it and try to analyze the results.

(a) We start by classifying the digits 0-1. Load the training data and test data using the function *get_digits()* in *load_data.py*. In order to get the 0/1 digits use: *x_train, y_train, x_test, y_test = get_digits(0, 1)*. This function use the 'Tensorflow' package. In order to use this package in the school read the instructions in cs-wiki, and watch this.
Run the SGD algorithm with batches of 5, 50, 100 and the GD algorithm over 150 iterations. **Make sure to change the labels to 1/-1**. Which eta did you used?

(b) **Draw** a graph of the training loss of each algorithm at each iteration and another graph of the test loss. Use the 0-1 loss for both losses.

(c) **Question:** The SGD with the batch tries to imitate the GD but with less samples as you can see. Analyze the results in the last bullet, and explain them. Explain the tradeoff between the batch size and the computational complexity and the accuracy of the subgradient in the learning process.

(d) Run through b and d again, but with eta = 0.1, 10. What can we learn from the learning rate results in this specific problem?
For what cases should we consider high learning/ low learning rate. You may consider the gradient of $f(x) = 10^8 \cdot |x|$ and $g(x) = 10^{-8} \cdot |x|$ at different points and its effect in GD when you answer the question.

(e) Try classifying another pairs of digits. Find a pair of Mnist digits that the GD or the SGD doesn't succeed very well on them and suggest an explanation for the bad results.

Logistic regression

21. (Bonus: 5 points) Create a file: "logistic_reg.py" that will contain functions gd and sgd according to the gradient that we found in the logistic regression section of the theoretical part. You should also implement the *test_error* function again. Note that in order to get classification you should define threshold, thus the declaration is: "*test_error(w, test_data, test_labels, threshold)*" - how did you chose threshold? (Hint: think of the meaning of 0.5).

22. (Bonus: 5 points) Repeat 20:b-c, with the gd, sgd functions from "logistic_reg.py". **This time the labels are 0/1**. Make sure to add a 1 at the end of each sample in order to work with affine hypothesis class instead of linear. Which eta did you used?

23. (Bonus: 5 points) Plot the ROC curve for the w you found by the GD in section b. What can you say about this curve?

The SGD is quite useful in real world problems. In the next exercise will use SGD on a non convex problem- such as a neural network!