# Algorithms for Optimizing Non-Differentiable Functions

Yizhi Mao
University of California, Berkeley
yizhi_mao@berkeley.edu
SID: 3032129270

Hanxiang Xie
University of California, Berkeley
xiehx1992@berkeley.edu
SID: 3032131844

Wei Jin
University of California, Berkeley
jinw@berkeley.edu
SID: 24875865

## Abstract

*Nondifferentiable function optimization is a problem frequently encountered in machine learning. Three algorithms that are commonly used are Subgradient method, Iterative Shrinkage Thresholding Algorithm (ISTA), and Fast Iterative Shrinkage Thresholding Algorithm (FISTA). To compare the convergence rate of the three algorithms, we implemented them to a practical classification problem with l1 norm added to the loss function. The empirical result shows that FISTA requires the fewest number of iterations to reach the converging value of loss function.*

## 1. Introduction

Least absolute shrinkage and selection operator (LASSO) performs both feature selection and regularization by adding L1 norm to the loss function. The L1 regularization term helps prevent model from overfitting and enhance model interpretability. However, the existence of L1 norm makes the loss function non-differentiable, rendering traditional gradient descent methods useless for optimization. This project studied and compared three algorithms for minimizing nondifferentiable objective functions: Subgradient Method, Iterative Shrinkage Thresholding Algorithm (ISTA), and Fast Iterative Shrinkage Thresholding Algorithm (FISTA). The three algorithms are implemented to a real-world classification problem that requires feature selection. Speed of convergence is compared based on the numbers of iteration required to reach the value of converged loss.

The report proceeds as follows. Section 2 describes the real-world data set for the classification problem, as well as how the data were pre-processed. In

section 3, we present the details of Subgradient method, ISTA and FISTA. Section 4 shows the results we achieved by implementing the three algorithms for classification in python. Section 5 discusses and presents the possible explanation for the result.

## 2. Data

The classification problem is based on the dataset from The 20Newsgroup corpus[1]. It contains approximately 20000 documents from 20 categories. For better illustration purpose, we picked a random subset of the dataset with 2 categories and their corresponding documents. Our goal is to classify the documents in the subset as one of the two categories, based their contents. Thus, the design matrix is a term-document matrix generated based on the documents, with row i representing document i and column j representing term j. Each matrix entry is the number of occurrence of term j in document i. The term-document matrix contains the most frequent 1000 terms among all the documents. The columns are normalized respectively. As a result, our design matrix, denoted as X, has 2027 rows and 1000 columns. X, along with the vector of labels, were used for implementing the following three methods.

## 3. Methods

To get the classification model, our goal is to minimize an objective function consists of cross-entropy and an L1 regularization term. Cross-entropy loss function is chosen because this is a classification problem. Also, feature selection would be achieved by the L1 regularization term in the objective function, which prevents the model from overfitting.

As a result, the objective function is

$$\min_w \sum_{i=1}^{n} \{y_i \ln s(X_i \cdot w) + (1 - y_i) \ln(1 - s(X_i \cdot w))\} + \lambda ||w||_1$$

The gradient of the cross-entropy is able to obtain due to its differentiability. However, the L1 norm at the end is not differentiable. Therefore, the ordinary gradient descent methods will not be helpful to solve this optimization problem.

Fortunately, there are three methods designed to solve this issue, which refer to subgradient method, ISTA and FISTA.

### 3.1 Subgradient Method

Subgradient method is a slow but simple algorithm used to solve optimization problem with non-differentiable objective function. The difference between ordinary gradient descent method is that it is not a descent method.

Subgradient method involves calculation of subgradient. The

subgradient for the objective function is derived to be:

$$g = -X^T(y - s(Xw)) + \lambda d$$

di is defined as:

$$d_i = \begin{cases} \text{sign}(w_i) & \text{if } w_i \neq 0 \\ d_i \in [-1, 1] & \text{if } w_i = 0 \end{cases}$$

The algorithm of subgradient method is as follows:

- Initialize $w^{(0)} = \vec{0}$
- Repeat

$$w^{(k)} = w^{(k-1)} - t \cdot g^{(k-1)}, k = 1, 2, 3, \ldots,$$

- Keep track of best iterate $w_{best}^{(k)}$ among $w^{(1)}, \ldots w^{(k)}$ i.e.

$$J(w_{best}^{(k)}) = \min_{i=1,\ldots k} J(w^{(k)})$$

For the step size t, it is fixed to be 0.05.

### 3.2 ISTA

Recently, ISTA have been introduced to solve a large number of convex unconstrained optimization problems. Despite the fact of its simplicity, ISTA is recognized as a slow method.

The algorithm of ISTA is as follows:

- Initialize $w^{(0)} = \vec{0}$
- Update the weight vector by the $p_L$ function defined as

$$p_L(w_k) = \tau_{\lambda t}\{w_k + tX^T(y - s(Xw_k))\}$$

where $\tau$ function is defined as

$$\tau_{\lambda t}(w)_i = (|w_i| - \lambda t)_+ sgn(w_i)$$

Furthermore, the optimal step size t for logistic regression equals one quarter of the largest eigenvalue of $X^TX$ [2]. In this example, step size t equals 0.00343.

- Repeat these two steps until convergence.

### 3.3 FISTA

Fast Iterative Shrinkage Thresholding Algorithm (FISTA) is an extension of ISTA with improved complexity. FISTA uses the same iterative shrinkage operator as ISTA, namely $p_L(\cdot)$. Unlike ISTA, which apply the $p_L(\cdot)$ to the previous point $w_{k-1}$, FISTA employs the shrinkage operator at a point $y_{k-1}$ which is a function of the previous two points $\{w_{k-1}, w_{k-2}\}$.

The detailed algorithms is as follows:

- Initialize $w^{(0)} = \vec{0}$
- Repeat the following three steps until convergence:

$$\text{(i)} \quad w_k = p_L(y_k)$$

$$\text{(ii)} \quad t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$$

$$\text{(iii)} \quad y_{k+1} = w_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(w_k - w_{k-1})$$

According to the algorithm, the computational effort for FISTA is the same as ISTA. The additional computation for FISTA (ii) and (iii) is marginal. It is theoretically shown that [4] with the improved complexity,

FISTA has a significantly faster global convergence rate than ISTA.

When FISTA is implemented in the classification problem, we initialized step size t equal to the optimal step size used in ISTA. Then t is updated in each iteration.

## 4. Result

For the purpose of comparability, we set the initialized weight to be the zero vector for all algorithms. Each algorithm is implemented with its corresponding learning rate. For each algorithm, the result includes the number of iterations for each algorithm to reach the converging loss value (approximately 320), as well as the path of loss function until it reaches the convergent value of cost.

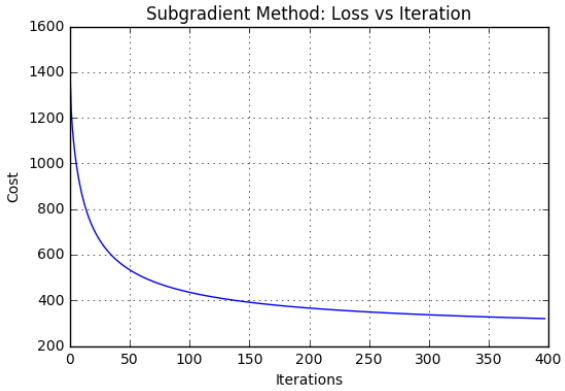


Figure 1: Cost for each iteration for subgradient method

Figure 1 shows the relationship between cost and the number of iterations for subgradient method. The cost drops dramatically within the first 50 iterations, and then flattens out for more iterations. The step size is constant, and it works well for this problem, since the cost function convergences to about 320 without overshooting problem.

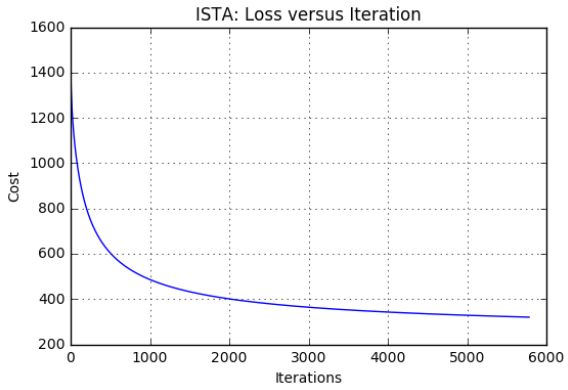


Figure 2: Cost for each iteration for ISTA

Figure 2 shows the relationship between cost and the number of iterations for ISTA. The cost drops dramatically within the first 1000 iterations, and then flattens out for more iterations. The number of steps required to converge is much greater than subgradient method.

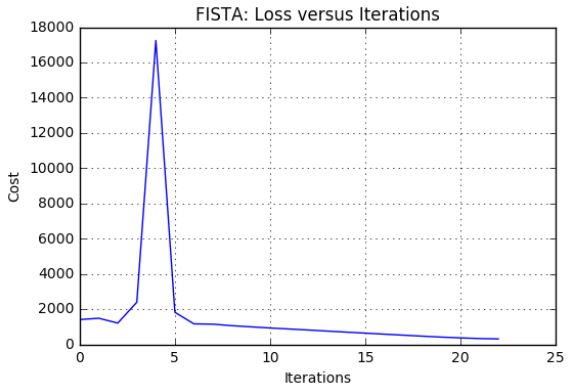


Figure 3: Cost for each iteration for FISTA

Figure 3 shows the relationship between cost and the number of iterations for FISTA. The cost first drops a bit but it has a big spike in the fourth iteration, and then converges quickly with less than 25 iterations.

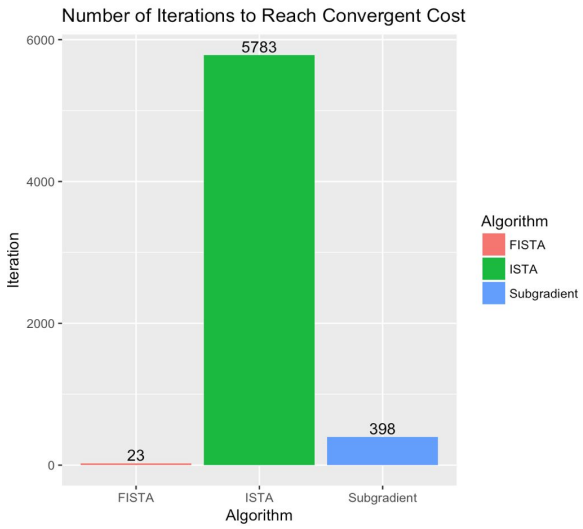


Figure 4: Number of iterations for FISTA, ISTA and Subgradient method

In order to compare different algorithm, the number of iterations required to converge is used as the criterion to compare efficiency of different algorithm. The stopping criterion is to get a cost value less than 320 in this case. Higher number of iterations means lower speed to converge. The number of iterations required for ISTA, subgradient method and FISTA is 5738, 398 and 23. FISTA is significantly faster than the other two algorithms. Although theoretically, the convergence rate for ISTA, subgradient method and FISTA is $O(\frac{1}{k})$, $O(\frac{1}{\sqrt{k}})$ and $O(\frac{1}{k^2})$ [3][4], in this problem, subgradient performs better than ISTA. The discrepancy comes from the fact that for this particular problem, fixed step size is used instead of diminishing step size.

## 5. Conclusion

This paper explores three methods to solve non-differentiable objective function optimization problems. To compare the performance of the algorithm, the number of iterations to reach a determined cost is used as the criterion. These algorithms were implemented in a classification problem using logistic regression with 20 newsgroup data set. The final result shows that FISTA beats the other two algorithms, followed by subgradient method and ISTA.

References

[1] The 20 Newsgroups data set: http://qwone.com/~jason/20Newsgroups/

[2] Chi, J. T., & Chi, E. C. (2014). Getting to the Bottom of Regression with Gradient Descent. From: http://jocelynchi.com/pdf/gettingtothebottom_gradient_descent_R_tutorial.pdf

[3] Tibshirani, R. (2012). Convex Optimization Course (Machine Learning 10-750), Lecture 7 scribing. From: https://www.cs.cmu.edu/~ggordon/10725-F12/scribes/10725_Lecture7.pdf

[4] Beck. A., & Teboulle. M. (2009). A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. In *Siam J. Imagine Sciences* (Vol. 2, No. 1, pp. 183–202).