Project Work in Optimization Methods/Optimization Techniques for Machine Learning

Mbakop Dimitri

January 2024

1 Assignment

1.1 Project Topic

Evaluating the performance of Gradient Descent and Newton's methods on Logistic Regression Problems

1.2 Projects goals

- Implement (in Python language, exploit `numpy` library) classes and functions allowing to load data to build an instance of l_2-regularized logistic regression problem, compute the loss, the gradient of the loss and the Hessian (see Lecture notes, Sec. 2.3).
- Implement gradient descent and Newton's method. Gradient descent should be equipped with an Armijo-type line search.
- Identify a small set of real-world instances ($\sim$ 6 datasets), to carry out experiments.
- Test the efficiency of the considered methods on this benchmark of problems, measuring the number of iterations and the runtime needed to reach solutions that are (approximately) stationary points. Also report the objective value at the obtained final solutions.
- **Bonus task**: Include in the comparison the conjugate gradient method that can be found in the `scipy.optimize` module.

2 Introduction

Logistic regression is a statistical model for binary classification, predicting the probability of an instance belonging to a particular category[5].

It utilizes the logistic function (sigmoid function) to transform outputs into probabilities between 0 and 1. Parameters are adjusted through optimization to maximize the likelihood of accurate predictions. This model is effective for binary outcomes and widely applied in fields like medicine and finance for tasks such as disease diagnosis and credit scoring[11].

So we have a dataset:

$$\mathcal{D} = \{(x^{(i)}, y^{(i)} \mid x^{(i)} \in \mathcal{X}, y^{(i)} \in \mathcal{Y}, i = 1, \ldots, n\},$$

where $\mathcal{X} \subseteq \mathbb{R}^p$ and $\mathcal{Y} = \{0, 1\}$ (*binary classification*).

The dataset represents a sampling from some distribution, where some relation f exists between pairs (x, y), such that $f(x) = y$.

The aim in machine learning is to construct, based on the pairs in $\mathcal{D}$, a function $\hat{f}$ that captures the essence of f, being able to accurately provide values of $\hat{y} = \hat{f}(x)$ for point x that are not present in the training set $\mathcal{D}$.

Training is typically modeled as an optimization problem (*empirical risk minimization*), where a *loss function* has to be minimized w.r.t. the parameters w of the model f. The usual form of training optimization problems is

$$\min_w L(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i; w), y_i)$$

$$\text{Log loss: } \ell(u, v) = -\left(u \log(v) - (1 - u) \log(1 - v)\right)$$

The goal is to minimize the error on the entire data distribution, not just the training set, to ensure generalization to unseen data. *Overfitting*, a common issue with expressive models, occurs when the model is excessively tailored to the training data, leading to poor performance on new, unseen data.

To address *overfitting*, a regularization term is typically introduced in the training problem, enhancing the generalization capability of the learning model. This results in an optimization problem with this form

$$\min_w L(w) + \Omega(w)$$

with

$$\Omega(w) = \|w\|_2^2 \text{ (quadratic regularization)}$$

2.1 Logistic Regression

The expression of the Logistic regression is,

$$\min_{w \in \mathbb{R}^p} L(w) + \lambda \Omega(w)$$

The logistic loss function with quadratic regularization is given by:

$$L(w; X, y) = -\frac{1}{n} \sum_{i=1}^{n} \left[y^{(i)} \log(\sigma(w^T x^{(i)})) + (1 - y^{(i)}) \log(1 - \sigma(w^T x^{(i)})) \right] + \lambda \|w\|_2^2$$

where:

- n is the number of samples.
- w is the weight vector.
- X is the feature matrix.
- $y^{(i)}$ is the true label for the i-th sample.
- $x^{(i)}$ is the i-th feature vector.
- $\sigma(z)$ is the sigmoid function defined as $\frac{1}{1+\exp(-z)}$.
- λ is the regularization parameter.

For problems of this form, an iterative algorithm is required to train the model; methods such as gradient descent , Newton's method or conjugate gradient are all viable options[5].

The gradient of $L(w) + \lambda\Omega(w)$ with respect to w is:

$$\nabla_w (L(w) + \lambda\Omega(w)) = \nabla_w L(w) + 2\lambda w$$

or

$$\nabla L(w; X, y) = -\frac{1}{n} \sum_{i=1}^{n} \left[\left(y^{(i)} - \sigma(w^T x^{(i)}) \right) x^{(i)} \right] + 2\lambda w$$

The Hessian matrix of $L(w) + \lambda\Omega(w)$ with respect to w is:

$$H(w) = \nabla_w^2 (L(w) + \lambda\Omega(w)) = \nabla_w^2 L(w) + 2\lambda I$$

or

$$H(w) = \nabla^2 L(w; X, y) = \frac{1}{n} \sum_{i=1}^{n} \sigma(w^T x^{(i)}) \left(1 - \sigma(w^T x^{(i)}) \right) x^{(i)} x^{(i)T} + 2\lambda I$$

where I is the identity matrix.

2.2 Gradient Descent with Armijo Step Size

Gradient Descent is an iterative optimization algorithm used to minimize a differentiable function. In the context of logistic regression, the objective is to minimize the logistic loss function $L(w)$[8]. The Armijo step size rule is employed to determine the step size at each iteration. Given a starting point $w^{(0)}$, the update rule for the next iteration is [9]:

$$w^{(k+1)} = w^{(k)} - \alpha_k \nabla L(w^{(k)})$$

where α_k is the step size determined by the Armijo Rule. The Armijo Rule involves selecting a small constant $0 < c < 1$ and a reduction factor $0 < \rho < 1$. The step size α_k is chosen as the largest α that satisfies the Armijo condition:

$$L(w^{(k)} - \alpha \nabla L(w^{(k)})) \leq L(w^{(k)}) - c\alpha \nabla L(w^{(k)})^T \nabla L(w^{(k)})$$

This condition ensures that the new point after the update results in a sufficient reduction in the objective function.

2.3 Newton's Method

Newton's Method is an iterative optimization algorithm that uses second-order information, specifically the Hessian matrix, to find the minimum of a function. For logistic regression, the update rule at each iteration is given by[10]:

$$w^{(k+1)} = w^{(k)} - \alpha_k (H(w^{(k)}))^{-1} \nabla L(w^{(k)})$$

where $H(w^{(k)})$ is the Hessian matrix of $L(w)$ with respect to w, and α_k is a step size.

2.4 Conjugate Gradient Method

The Conjugate Gradient method is an iterative optimization algorithm that combines ideas from both gradient descent and Newton's method. It ensures conjugacy between search directions, leading to efficient convergence[6]. The update rule for logistic regression can be expressed as:

$$w^{(k+1)} = w^{(k)} + \beta_k d^{(k)}$$

where $d^{(k)}$ is the conjugate direction and β_k is the step size.

3 Experiments

3.1 Datasets

We conducted our experiments on five diverse datasets to evaluate the performance of the logistic regression model:

Breast Cancer Dataset

The Breast Cancer dataset consists of features computed from digitized images of fine needle aspirates (FNA) of breast masses. The task is to classify whether a tumor is malignant or benign[1].

Ionosphere Dataset

The Ionosphere dataset involves radar data collected from an ionospheric sounder. The objective is to determine whether a signal passes through the ionosphere or not, making it a binary classification problem[7].

Diabetes Dataset

The Diabetes dataset contains measurements related to diabetes patients, such as age, BMI, blood pressure, and six blood serum measurements. The objective is to predict based on diagnostic measurements whether a patient has diabetes[2].

Wine Quality Dataset

The Wine Quality dataset comprises chemical properties of red and white wines, and the goal is to predict the quality of the wine based on these features[3].

Rice (Cammeo and Osmancik) Dataset

The Rice dataset, including data from Cammeo and Osmancik varieties, focuses on various agricultural parameters of rice plants. The task involves predicting the class of the rice (Cammeo or Osmanik) using the characteristics related to rice cultivation[4].

3.2 Metrics

We tested each optimization method, including Gradient Descent with Armijo Step Size, Newton's Method, and Gradient Conjugate, on the five diverse datasets. We will compare the results based on the following metrics:

1. **Last Cost Value:** The final value of the cost function after the optimization process. It provides insight into how well the chosen method minimized the objective function.

2. **Runtime:** The time taken for the optimization process to complete. It measures the efficiency of each method in terms of computational speed.

3. **Iterations:** The number of iterations required for the optimization process to converge. It indicates the convergence speed and efficiency of the algorithm.

4. **Accuracy:** The classification accuracy of the logistic regression model on the test dataset. This metric measures the predictive performance of the model.

3.3 Results

We show the results on the table 1

Dataset	Optimization Method			
	Metric	**Gradient Descent**	**Newton's Method**	**Conjugate Gradient**
Breast Cancer	Last Cost Value	0.1419	0.6916	0.0701
	Runtime (seconds)	0.0187	0.0026	0.0159
	Iterations	49	2	30
	Accuracy (%)	97.37	94.74	98.25
Ionosphere	Last Cost Value	0.3231	0.6851	0.2017
	Runtime (seconds)	0.0486	0.0120	0.1408
	Iterations	70	8	48
	Accuracy (%)	88.73	88.73	87.32
Diabetes	Last Cost Value	0.5600	0.6929	0.5273
	Runtime (seconds)	0.0099	0.0055	0.0061
	Iterations	38	1	12
	Accuracy (%)	70.13	68.83	68.83
Wine Quality	Last Cost Value	0.5969	0.6931	0.5542
	Runtime (seconds)	0.0857	0.2072	0.1129
	Iterations	33	1	26
	Accuracy (%)	68.54	68.46	69.15
Rice (Cammeo and Osmancik)	Last Cost Value	0.2454	0.6930	0.1996
	Runtime (seconds)	0.0274	0.0235	0.0251
	Iterations	49	1	23
	Accuracy (%)	92.13	92.78	92.39

Table 1: Experimental Results for Logistic Regression Optimization Methods

3.3.1 Comparison between Gradient Descent and Newton's Method

Newton's method exhibits superior convergence speed and requires fewer iterations compared to gradient method. However, it faces challenges in surpassing gradient descent in terms of accuracy and minimizing the loss value. Additionally, Newton's method incurs significant computational costs, making it demanding in terms of computational resources.

3.3.2 Comparison among Three Methods

The conjugate gradient method demonstrates greater efficiency than gradient methods in minimizing loss values and is frequently more accurate than gradient descent. While it is slower than Newton's method, it often outpaces gradient methods and is less computationally demanding. This positions the conjugate gradient method as a well-balanced alternative, striking a favorable compromise between the gradient conjugate and Newton methods.

4 Conclusion

Selecting the most appropriate optimization method for logistic regression involves a trade-off between convergence speed, computational efficiency, and accuracy. The Gradient Conjugate method emerges as a robust choice, striking a balance between the advantages of Gradient Descent and Newton's Method. However, the final decision should consider the specific characteristics of the dataset and the available computational resources. As machine learning practitioners, understanding the strengths and considerations of each optimization method is crucial for making informed decisions when implementing logistic regression models.

The code can be found here.

References

[1] Abien Fred M Agarap. "On breast cancer detection: an application of machine learning algorithms on the wisconsin diagnostic dataset". In: *Proceedings of the 2nd international conference on machine learning and soft computing*. 2018, pp. 5–9.

[2] Dilip Kumar Choubey et al. "Classification of Pima indian diabetes dataset using naive bayes with genetic algorithm as an attribute selection". In: *Communication and computing systems: proceedings of the international conference on communication and computing system (ICCCS 2016)*. 2017, pp. 451–455.

[3] Parneeta Dhaliwal, Suyash Sharma, and Lakshay Chauhan. "Detailed study of wine dataset and its optimization". In: *Int. J. Intell. Syst. Appl.(IJISA)* 14.5 (2022), pp. 35–46.

[4] Umit Ilhan et al. "Classification of osmancik and cammeo rice varieties using deep neural networks". In: *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*. IEEE. 2021, pp. 587–590.

[5] Michael P LaValley. "Logistic regression". In: *Circulation* 117.18 (2008), pp. 2395–2399.

[6] John L Nazareth. "Conjugate gradient method". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 1.3 (2009), pp. 348–353.

[7] Alessio Pignalberi, Marco Pietrella, and Michael Pezzopane. "Towards a real-time description of the ionosphere: a comparison between international reference ionosphere (IRI) and IRI real-time assimilative mapping (IRTAM) models". In: *Atmosphere* 12.8 (2021), p. 1003.

[8] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *arXiv preprint arXiv:1609.04747* (2016).

[9] Zhenjun Shi and Shengquan Wang. "Modified nonmonotone Armijo line search for descent method". In: *Numerical Algorithms* 57 (2011), pp. 1–25.

[10] Yong Wang. "Gauss–newton method". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 4.4 (2012), pp. 415–420.

[11] Xiaonan Zou et al. "Logistic regression model optimization and case analysis". In: *2019 IEEE 7th international conference on computer science and network technology (ICCSNT)*. IEEE. 2019, pp. 135–139.