

---

# Logistic Regression with $L_2$ Regularization

---

Adrian Guthals, David Larson, Jason Yao

CSE 250B: Project #1

University of California, San Diego

## Abstract

The Abstract paragraph should be indented 1/2 inch (3 picas) on both left and right-hand margins. Use 10 point type, with a vertical spacing of 11 points. Two line spaces precede the Abstract. The Abstract must be limited to one paragraph.

## 1 INTRODUCTION

This paper will attempt to recreate the results by [1].

Highlight what Logistic Regression, why is it used and where is it used (real-world applications). Also mention potential pitfalls of the method (e.g. overfitting).

## 2 DESIGN AND ANALYSIS OF ALGORITHMS

Stochastic gradient descent (SGD) and Limited-memory Broyden-Fletcher-Goldfarb-Shannon (L-BFGS) were each implemented in Matlab to maximize the total conditional log likelihood of each training data set. In brief, SGD incorporated a fixed learning rate  $\lambda$  to control the change in log likelihood, which was averaged over random mini-batches of size  $\kappa$ . To control over-fitting, logistic regression was used and change in the objective function was evaluated on a separate validation dataset containing 30% of all training examples selected at random. Convergence was reached when change in the objective function was less than  $\omega$  or the total number of epochs was greater than  $\varepsilon$  (which ever came first). (Add brief overview of L-BFGS and cite MinFunc's implementation)

For each of these algorithms, the input training data is formatted as a set of  $n$  examples  $x_1 \dots x_n$  where each  $x_j$  is a real-valued vector of  $d$  features. Each  $x_i$  is correlated to a binary (Bernoulli) outcome  $y_i$  by a global parameter vector  $\beta$  of length  $d + 1$ . We assume this correlation follows the model below where  $x_{i0} = 1$  for all  $i$ .

$$p_i = p(y_i | x_i; \beta) = \frac{1}{1 + \exp - (\sum_{j=0}^d \beta_j x_{ij})} \quad (1)$$

## 2.1 STOCHASTIC GRADIENT DESCENT

Our SGD implementation first randomized the order of input examples to avoid repeated computation of random numbers and partitioned the input data into  $x_1 \dots x_k$  *training* examples and  $x_{k+1} \dots x_n$  *validation* examples. Then sequential mini-batches of size  $\kappa < k$  taken from the training set were used to update the parameter vector  $\beta$  (initialized to all zero values) by the following equation. The constant  $\mu$  quantifies the trade-off between maximizing likelihood and minimizing parameter values for  $L_2$  Regularization.

$$\beta = \beta + \frac{\lambda}{\kappa} [-2\mu\beta + \sum_{i=1}^{\kappa} (y_i - p_i) x_i] \quad (2)$$

After each update of  $\beta$ , absolute change in the objective  $\hat{\beta}$  was computed over all validation examples with the following function.

$$\hat{\beta} = \mu \|\beta\|_2 + \sum_{i=k+1}^n -\log(p_i^{y_i} (1 - p_i)^{1-y_i}) \quad (3)$$

Convergence was reached when change in the objective reached a value less than  $\omega$ . Convergence was also reached if the total number of epochs was greater than  $\varepsilon$ .

## 2.2 L2 REGULARIZATION

$$\hat{B} = \operatorname{argmax}_{\beta} LCL - \mu \|\beta\|_2^2 \quad (4)$$

where  $\|\beta\|_2^2$  is the  $L_2$  norm of the parameter vector.

$$\frac{\partial}{\partial \beta_j} LCL = \sum_i (y_i - p_i) x_{ij} \quad (5)$$

## 2.3 LIMITED-MEMORY BFGS

Limited-memory Broyden-Fletcher-Goldfarb-Shannon (L-BFGS) is a quasi-Newton optimization method used to find local extrema.

## 3 DESIGN OF EXPERIMENTS

Design of experiments.

### 3.1 HYPERPARAMETERS

#### 3.1.1 Learning Rate

How did we determine  $\lambda$  (the learning rate).

#### 3.1.2 Regularization

How did we determine  $\mu$  (the regularization constant).

#### 3.1.3 SGD Subset

How did we determine (or why did we choose) the length of subsets used by the SGD (1 row at a time, 2 at a time, etc.)

## 4 RESULTS OF EXPERIMENTS

Results of experiments. Comparison of methods and data sets.

### 4.1 USPS-N

How did SGD and L-BFGS perform on the USPS-N data sets.

### 4.2 WEB

How did SGD and L-BFGS perform on the Web data sets.

#### 4.2.1 Figures

Figure 1: Sample Figure Caption

## 5 FINDINGS AND LESSONS LEARNED

Findings and lessons learned.

Table 1: Sample Table Title

PART	DESCRIPTION
Dendrite	Input terminal
Axon	Output terminal
Soma	Cell body (contains cell nucleus)

## References

- [1] N. Ding and S. Vishwanathan, “t-logistic regression,” in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., 2010, pp. 514–522.