CS 229 Course Project, Fall 2005 (Dec 16, 2005)

L1 Regularized Logistic Regression

Tejas Rakshe S. and Ashish Kumar

Abstract

We implemented subgradient descent - like method for L1 regularized logistic regression and nonlinear conjugate gradient method for Huber loss function regularized logistic regression. We investigated various aspects of these algorithms and used them on various datasets (obtained from the University of California at Irvine repository and CS229 class).

Purpose

To implement logistic regression with regularization using L1 and L1-like norms with different implementation strategies and investigate their properties.

Notation

The function to be minimized is of the form $f(\theta) + \lambda ||\theta||_n$

$f(\theta)$ is the *negative* of log likelihood function of logistic regression.

$||\theta||_n$ is the regularization term. n=1 for L1 norm. λ (≥ 0) is the weight of regularization term. The noise (if added) is such that *p* fraction of the training labels were flipped.

Methods

(A) Subgradient descent – like algorithm

Motivation

L1 norm is not differentiable at zero and usual gradient descent can oscillate when one of the parameters is close to zero. To overcome the problem, we modify the update rule for the weights θ.

Update rule for gradient descent:

For a small positive ε (~ 0.01),

If $|\theta_i| > \varepsilon$, perform usual update [class notes, cs229]

If $|\theta_i| \leq \varepsilon$ and

$|\delta(f) / \delta(\theta_i)| \leq \lambda$, then don't change θ_i.

If $|\theta_i| \leq \varepsilon$ and

if $\delta(f) / \delta(\theta_i) > \lambda$, then $\theta_i := \theta_i + \alpha ((\delta(f) / \delta(\theta_i)) - \lambda)$

if $\delta(f) / \delta(\theta_i) < -\lambda$, then $\theta_i := \theta_i + \alpha ((\delta(f) / \delta(\theta_i)) + \lambda)$

Step size (α):

We vary it as $\alpha(\text{iteration}) = \alpha_0 / \text{sqrt}(\text{iteration})$ [Boyd 2003]. Advantage - the step size slowly decreases, making the algorithm stable (ie, it converges and doesn't oscillate). Disadvantage - if the initial guess is too far from the actual solution, we may not be able to reach the actual solution, since the step size diminishes.

(B) Nonlinear Conjugate Gradient (with Newton-Raphson and Fletcher-Reeves) method for Huber loss regularization

Motivation for Huber loss regularization: Why is it better than L1 or L2?
The Huber loss function approximates L1 norm, and also has a desirable property of being differentiable everywhere, unlike L1 norm. It's a quadratic function near the origin and linear otherwise, such that the function is continuous and differentiable. It behaves like an L2 norm regularization for smaller residuals and like an L1 norm regularization for larger residues. It's considered robust, due to the fact that it does not heavily penalize higher residuals at the cost of lower residuals (as L2 norm does). At the same time it doesn't insist on driving residuals to zero, at the cost of leaving some residues very high (as L1 norm does) [Boyd 2004]. Notation for the Huber loss function is same as that used in [Boyd 2004].

What is Nonlinear Conjugate Gradient (with Newton-Raphson and Fletcher-Reeves) ? : [Shewchuk, 1994] Please refer to the paper for the details of the algorithm. However we modified the algorithm slightly to make it more robust.

Psuedocode in its original form:

$i \Leftarrow 0$
$k \Leftarrow 0$
$r \Leftarrow -f'(x)$
$d \Leftarrow r$
$\delta_{new} \Leftarrow r^T r$
$\delta_0 \Leftarrow \delta_{new}$
While $i < i_{max}$ and $\delta_{new} > \varepsilon^2 \delta_0$ do
    $j \Leftarrow 0$
    $\delta_d \Leftarrow d^T d$
    Do
        $\alpha \Leftarrow -\frac{[f'(x)]^T d}{d^T f''(x) d}$
        $x \Leftarrow x + \alpha d$
        $j \Leftarrow j + 1$
    while $j < j_{max}$ and $\alpha^2 \delta_d > \epsilon^2$
    $r \Leftarrow -f'(x)$
    $\delta_{old} \Leftarrow \delta_{new}$
    $\delta_{new} \Leftarrow r^T r$
    $\beta \Leftarrow \frac{\delta_{new}}{\delta_{old}}$
    $d \Leftarrow r + \beta d$
    $k \Leftarrow k + 1$
    If $k = n$ or $r^T d \leq 0$
        $d \Leftarrow r$
        $k \Leftarrow 0$
    $i \Leftarrow i + 1$

Modified form:

α Update: We use $\frac{ab}{b^2 + \varepsilon^2}$ instead of $\frac{a}{b}$ for the α update step, for a small ε, to eliminate divide-by-zero error (in case initial guess is too far away from the actual solution). This makes the algorithm stable. The problem of divide-by-zero error [Shewchuk, 1994] is common for conjugate gradient method.

Initial point: We use a special scheme for choosing the initial point. We solve the problem without regularization by gradient descent and use the solution as the initial guess. Unregularized gradient descent problem is computationally efficient to solve, and gives a solution reasonably close to the actual solution.

Experiments
We used validation scheme with 70% training and 30% testing data, with p=0 and p=0.1 or 0.2 (ie, with label flipping noise).

Datasets We used the algorithm on 3 different datasets. (1) Homework 1 data for CS229, (2) Voting patterns dataset (UC Irvine) (3) Breast cancer diagnosis data (UC Irvine) Please see the web links for details of the datasets]
(2) - http://www.ics.uci.edu/~mlearn/databases/voting-records/
(3) - http://www.ics.uci.edu/~mlearn/databases/breast-cancer-wisconsin/

Results
Figure 1 shows each one of 3 datasets for subgradient-like algorithm. Graphs (a), (b) and (c) are for one of three datasets each. In each graph there are plots for training and misclassification errors, with and without label-flipping noise. Figure 2 shows the same with non-linear CG algorithm with Huber loss.

Discussion
We implemented L1 and L1-like regularized logistic regression successfully with two different methods, and demonstrated it's application on three datasets. Each of the datasets comes from a different source and has different characteristics. We can see a general trend in all graphs that regularization produces better test results with noisy training data. In some datasets, there is a clear optimum for the regularization parameter, where the test data error is lowest. Also we note Huber loss is more robust than L1 norm to outliers as can be seen by the improvement of test error in all three cases with its respective training error for noisy data.

Work in Progress
1) Implementation of SMO-like algorithm
2) Use of exponential priors (Joshua Goodman's approach)
3) Convergence analysis for all the algorithms
4) Computational efficiency analysis and comparison

References
1. Ng, Andrew Y., "Feature selection, L1 vs. L2 regularization and rotational invariance", Proceedings of the 21st international conference on Machine Learning, 2004
2. Shewchuk, Jonathan R., "An Introduction to the Conjugate Gradient Method Without the Agonizing Pain", School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, 1994
3. Stephan Boyd, Lin Xiao and Almir Mutapcic, "Subgradient Methods", Notes for EE392o, Stanford University, 2003
4. Stephan Boyd and Lieven Vandenberghe, "Convex Optimization", Cambridge University Press, 2004.

Acknowledgements
We thank *Prof. Andrew Ng*, *Rajat Raina*, *Ashutosh Saxena* and *Rion Snow* of CS department, Stanford University, for the invaluable help we received from them regarding various aspects of the project. We are also grateful to people at the University of California at Irvine for making several datasets available for public use in their repository.

Effect of Regularization on Misclassification using L1 Norm Criteria

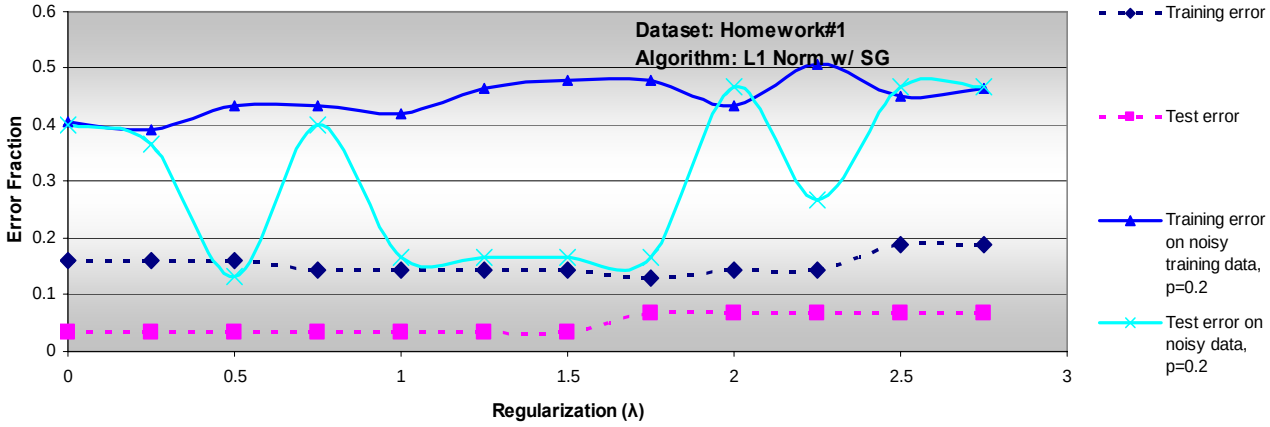

Fig 1 (a)

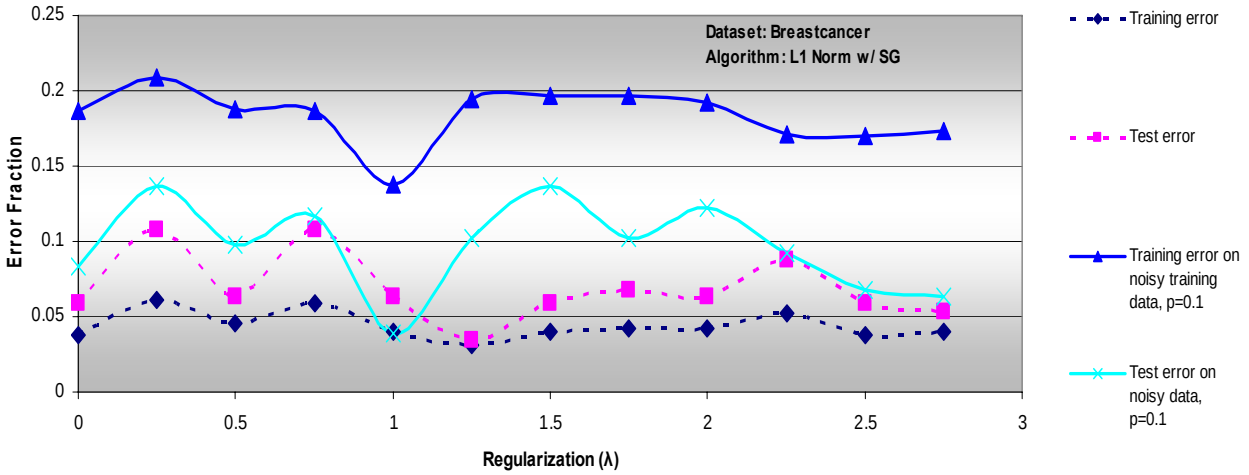

Fig 1 (b)

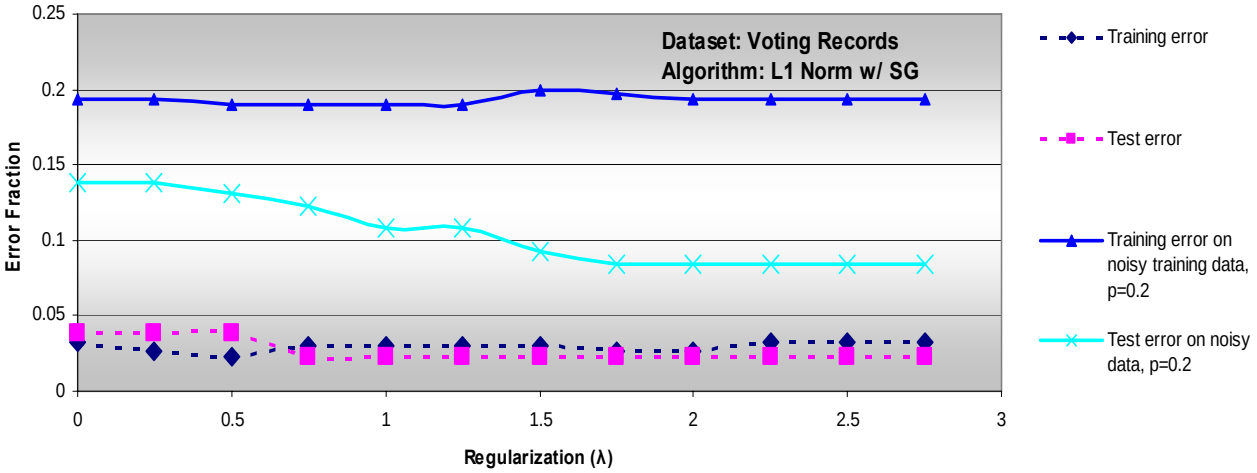

Fig 1(c)

Effect of Regularization and Noise on Misclassification using Huber Loss Criteria

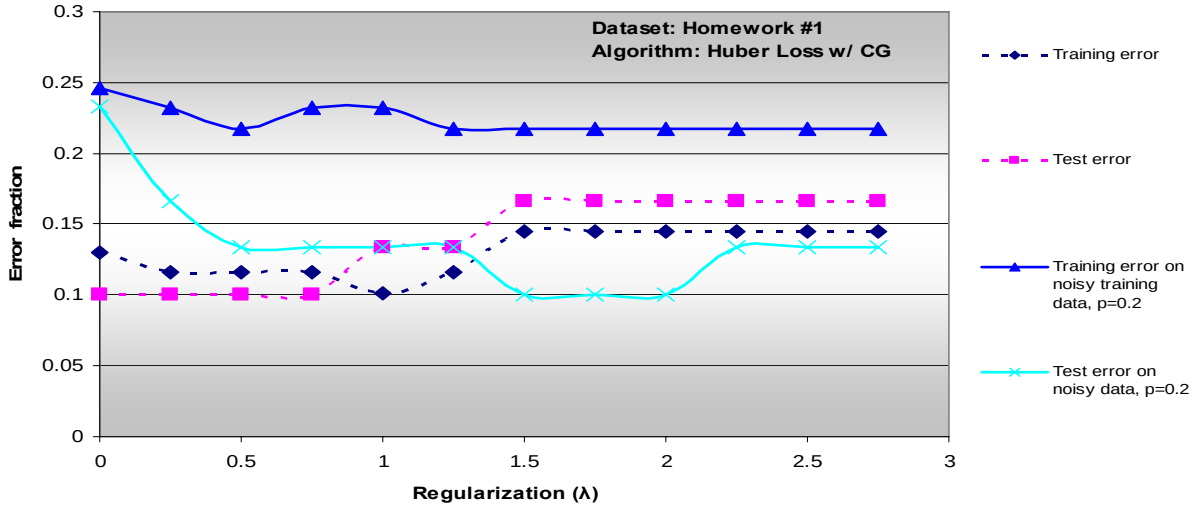

Fig 2 (a)

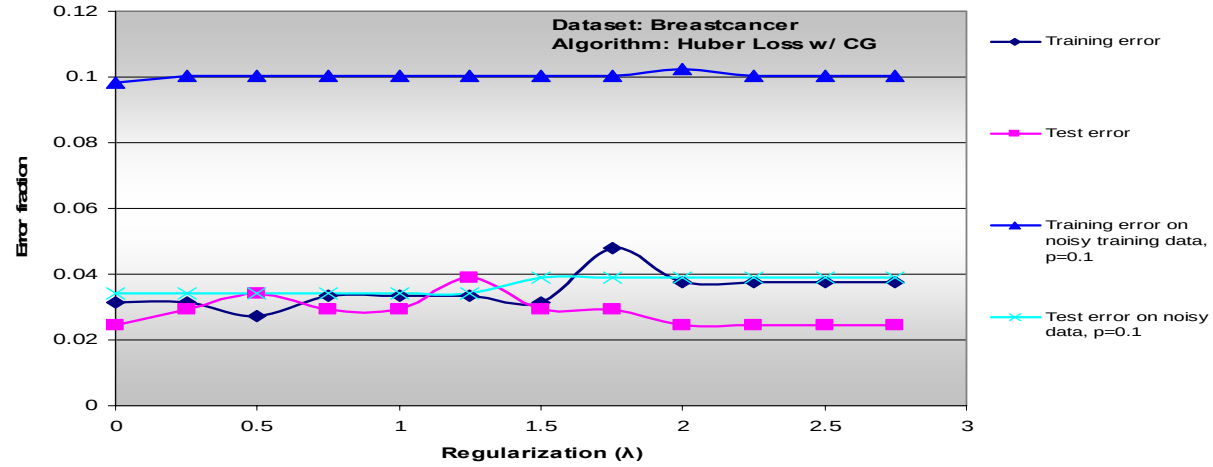

Fig 2 (b)

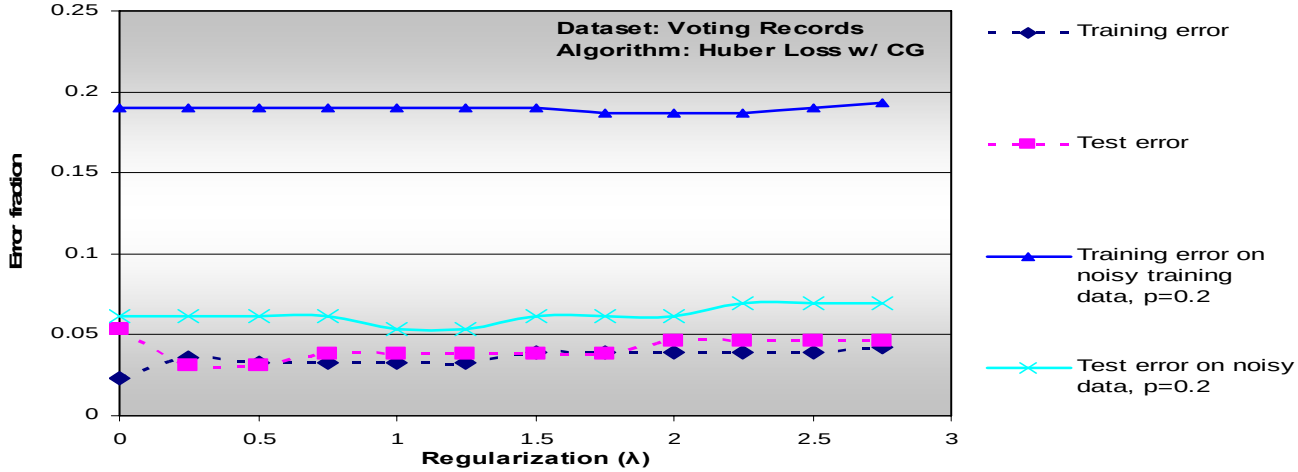

Fig 2 (c)