Project update

Timeline as of Nov. 17th

Nov 10th – completion of ridge regression code w/ least squares loss

Nov 17th – completion of lasso code w/ least squares loss

Nov 22nd – completion of lasso and ridge regression code w/ hinge loss

Nov 27th – completion of neural network code

Dec 1st – completion of rough first draft of final project report (assuming no progress of generative model)

Dec 8th – completion of generative neural network code

Dec 10th – completion of final draft of final project report

Project Status

Originally, I set a very aggressive timeline in hopes of developing a generative neural network which would have been above and beyond the scope of this class. This timeline was met or exceeded up until Nov 22nd. To save time and effort, I elected to postpone neural network code development until an example code was available through the assignments. Since this code was released as of November 30th, I have only developed a rough code for the MNIST dataset which is functional for 2-class binary classification but requires looping, verification, and testing. I certainly do not have a rough draft completed. Given the current status of my project and the pace of the class material, I will pursue a more-conventional submission: analysis of various techniques for binary classification without the development of a generative neural network. This adequately satisfies the project requirements and I am very well-positioned to complete that goal.

Since the last update I have completed l1-regulatized and l2-regulatrized hinge loss codes for 0th-degree, 1st-degree, and 2nd-degree fits. As previously stated, I have also completed a very preliminary neural network code – it classifies for only 2 classes ("1" or "not 1") and I've yet to reach the "testing" phase at all. The hinge loss codes, on the other hand, are complete.

I used two sets of data: the MNIST "training" dataset of 60000 images for training and the MNIST "testing" dataset of 10000 images for testing. The features used were each pixel in the image (784 pixels total). All reported error rates are the trained model's error rate when predicting testing dataset classes (the testing dataset was independent from the training dataset). Further work may involve rerunning the codes for different training/validation/testing schemes.

Results

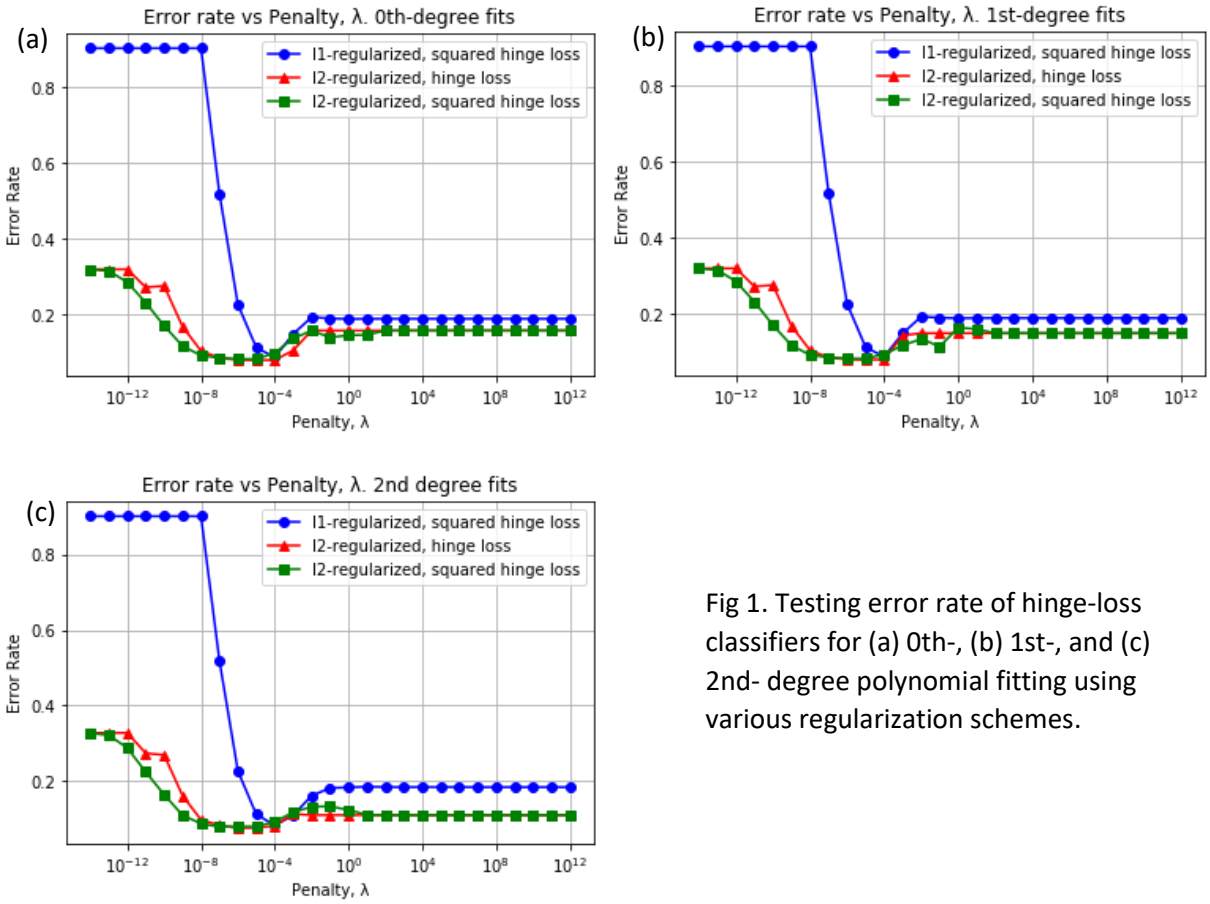

Fig 1. Testing error rate of hinge-loss classifiers for (a) 0th-, (b) 1st-, and (c) 2nd- degree polynomial fitting using various regularization schemes.

Observations

The hinge loss work is effectively finished and there are a number of noteworthy observations. First, its optimal error rate is the best that I've seen among all the classifiers that I've developed thus far.

Second, there is nearly no change in accuracy as polynomial fitting degree is increased from 0 up to 2, no matter if the hinge loss is l1-regularized or l2-regularized. The difference from fitting degree is only apparent for large penalty fits. This consistency across fitting degree mirrors the degree consistency found in my l1-regularized least-squares loss classifiers. So far, only the l2-regularized least-squares loss classifier showed noticeable improvement in error rate when fitting to higher orders.

The 0th-degree l2-regularized squared-hinge-loss classifier appears to be my best classifier so far. It is more accurate than the classifiers implementing other loss functions and it is much faster than the higher-order hinge loss classifiers.

Updated Timeline as of Dec. 1st

I will pursue the "backup" project proposal lined out in the proposal and first update submissions. The timeline has been revised to account for the neural network code status. Ideally, I would like to create a neural network of three layers (2 hidden layers of nodes, 3 sets of weights) or more to gain some experience in expanding a network out.

Nov 10th – completion of ridge regression code w/ least squares loss

Nov 17th – completion of lasso code w/ least squares loss

Nov 22nd – completion of lasso and ridge regression code w/ hinge loss

Dec 1st – complete rough code of neural network binary classifier (unlooped)

Dec 3rd – complete 2-layer neural network code

Dec 5th – complete 3-layer neural network code

Dec 10th – completion of final draft of final project report