

### Homework 3 (Due Monday 3/4)

#### CS534 Machine Learning, Spring 2019

This homework will explore validation and model selection procedures using regularized logistic regression models and nested cross validation.

#### Problem 1 - Cross validation (30 points)

This elastic-net regularized logistic regression model is derived by minimizing the negative log likelihood function for samples  $(x_i, g_i), g_i \in \{1, 2\}$

$$\max_{\beta, \beta_0} \ell(\beta, \beta_0) = \max_{\beta, \beta_0} \frac{1}{N} \sum_{i=1}^N \{I(g_i-1) \log p(x_i) + I(g_i-2) \log (1-p(x_i))\} - \lambda P_\alpha(\beta),$$

where the regularization penalty  $P_\alpha(\beta) = (1-\alpha)\frac{1}{2}\|\beta\|_2^2 + \alpha\|\beta\|_1$  is a function of the free parameters  $\lambda, \alpha$ , and where the indicator function  $I(0) = 1$  and zero elsewhere. The weight  $\lambda$  determines the magnitude of regularization, and the mixing parameter  $\alpha$  determines the proportion of the penalty allocated to the 1 and 2 norms. The mixing parameter  $\alpha$  is often not chosen through cross validation but just set using intuition or experience, where the weight  $\lambda$  is typically determined through cross validation.

In this problem you will use a built-in function for elastic-net logistic regression. The mixing parameter will be fixed  $\alpha = 0.95$  and you will search for the optimal regularization weight lambda in the range  $\lambda \in [0, 100]$ . These are often spaced on a logarithmic scale at 100 or more intervals.

Perform a ***nested*** cross validation using five folds to determine the optimal regularization weight and report test error. In each step, 4/5 of the data will be used for training and validation, and 1/5 will be used to report test error.

**1.a. Validation diagram (10 points)** Draw a diagram of the cross validation where you indicate which samples are in training, testing, and validation in each of the 5 nested folds. Depict both the inner and outer loops of the nested CV. Indicate clearly how the data is segmented, and the fraction of the data used in each segment.

**1.b. Model selection (10 points)** For each outer CV fold, generate a plot of the classification error on the validation set as a function of the sequence of  $\lambda$  values (five plots total). Use the inner CV folds to calculate standard deviations of the error  $\sigma_E(\lambda)$  at each  $\lambda$ , as well as the mean error  $\mu_E(\lambda)$ .

Choose the optimal lambda  $\lambda^*$  as the largest  $\lambda$  within 1-standard deviation of the minimum average error

$$\lambda_{min} = \underset{\lambda}{\operatorname{argmin}} \mu_E(\lambda),$$

$$\lambda^* = \max\{\lambda\} \text{ subject to } \lambda \leq \mu_E(\lambda_{min}) + \sigma_E(\lambda_{min})$$

In each plot, indicate  $\mu_E(\lambda)$  with a solid black line, and the intervals of variance  $\mu_E(\lambda) \pm \sigma_E(\lambda)$  in red,  $\mu_E(\lambda_{min})$  as a blue point and  $\mu_E(\lambda^*)$  as a green point.

**1.c. Test error (10 points)** Generate a box plot of the five test errors generated by cross-validation. Display the validation errors  $\mu_E(\lambda^*)$  in two additional boxplots in the same graph (there are 20 training errors and 20 validation errors). Compare and discuss the errors.

## Problem 2 (optional) - Elastic net logistic regression (25 points)

Implement the elastic net regression algorithm using the soft-thresholding and iterative reweighted least-squares approach described in [Friedman 2010](#). This problem can be solved by optimizing the penalized negative log likelihood

$$\min \ell(\beta, \beta_0) = \frac{1}{N} \sum_{i=1}^N \{I(g_i - 1) \log p(x_i) + I(g_i - 2) \log (1 - p(x_i))\} - \lambda P_\alpha(\beta),$$

where the indicator function  $I(0) = 1$  and zero elsewhere, and  $p(x_i)$  represents the class probability

$$\Pr(G = 1 | X = x_i) = p(x_i) = \frac{1}{1 + e^{-(\beta_0 + x_i^T \beta)}}.$$

A quadratic approximation of this likelihood  $\ell_Q$  can be used to transform this problem into a weighted least-squares problem with weights  $w_i$  and response  $z_i$  (see Equation 10 in Friedman paper)

$$\ell_Q(\beta, \beta_0) = -\frac{1}{2N} \sum_{i=1}^N w_i (z_i - \beta_0 - x_i^T \beta)^2.$$

where

$$z_i = \tilde{\beta}_0 + x_i^T \tilde{\beta} + \frac{y_i - \tilde{p}(x_i)}{w_i},$$
$$w_i = \tilde{p}(x_i)(1 - \tilde{p}(x_i)).$$

### 2.a. Solution path

Implement this algorithm using the mixing parameter  $\alpha = 0.95$  and plot the solution path  $\beta(\lambda)$  for a sequence of 1000 logarithmically spaced  $\lambda$  in the range  $\lambda \in [1, 100]$ . Label each feature  $j$  using text on the plot at the point where  $\beta_j$  enters the model (when this model coefficient becomes non-zero).

## Notes

- Understanding the approach at a high-level is important. You are going to solve the quadratic approximation repeatedly using equation 10 from the paper, and at each iteration you will recalculate  $z_i, w_i$  to update the quadratic approximation  $\ell_Q$ .
- You will generate a solution for each value of  $\lambda$ . A fixed number of iterations is the easiest approach for deciding when each problem is solved (50 worked for me).
- To accelerate convergence you can use a warm-starting technique. Start with the largest  $\lambda$  which will have the fewest non-zero coefficients. Use the solution of this regularization level as the start for the next smallest lambda, etc.
- This problem has some numerical issues that need to be addressed. The weights,  $w_i$  can shrink to zero and cause NaN values where used in division. The parameters  $\beta$  will continue to grow in an attempt to push the probabilities  $p_i$  to zero or one. For this reason, you can clamp these to 0, 1 once they are within a reasonable distance (say  $1e-5$ ). The solutions may also become unstable when the regularization gets too small (this will be apparent in the solution path plot).