Ecs171 HW3

Spencer Pham

December 3, 2019

Problem 1.

Lasso regression is very similar to ridge regression in that it adds a bias and reduces variance to the regression. It does so by adding a penalty to each weight, called λ. However, instead of squaring each weight like ridge regression, lasso takes the absolute value of each weight and multiple that by the penalty: $min[(y - wX)^2 + \lambda\Sigma^n|w|]$. Furthermore, something significant that lasso regression does is that it actually removes features, unlike ridge regression, which only reduces the impact of a feature (but never zeros it out).

Optimal parameter val: 0.000494
Num non-zero coeff: 131
Gen Error: 5.26%

Problem 2.

My methodology is to use the model from problem 1 (lasso regression that was cross-validated using 5 K-folds). Then, the gene expressions (called X), was bootstrapped 1000 times. Each bootstrap was resampled with the same number of samples (195), with replacement. For each bootstrap, the mean of X was calculated and used to predicted the growth rate (y). In total, 1000 y's were predicted and inserted into a list called $y\_predicts$. Finally, we use that y dataset and plug it into the confidence interval equation, which is $(\bar{X} \pm 1.96\frac{\sigma}{\sqrt{n}})$, where σ, n are the standard deviation and size of X respectively.

Assumptions:

1) The samples were drawn randomly and independently from a true population, which allow us to assume a normal distribution.

2) Since the sample size is greater than 25 (from bootstrapping), the difference between using a z-score and t-score should be negligible, so a z-score is used in this case.

3) We assume a 95% confidence interval is large enough to accommodate outliers and noise, while still giving a precise interval of the predicted value.

Problem 3.

[0.392785 0.394853], assuming that we're looking for a confidence interval of 95%.

Problem 4.

Since I'm reusing the same features that were selected from the regularized regression technique from P1, there still exists 131 non-zero coefficients. For each y being classified (strain, med, env_pert, gene_pert), the ROC AUC were (.8, .79, .77, .76) and PR AUC were (.4, .13, .31, .3) for each respective y.

Problem 5.

Composite classifier: AUC = 0.8, AUPRC = 0.23
Two individual classifiers:
medium: AUC = 0.79, AUPRC = 0.13
environmental_pertubation: AUC = 0.77, AUPRC = 0.31

The composite classifier appears to be slightly better (by .01 and .03 respectively) for the AUC, but it is in between for AUPRC. Hence, it likely would not make a significant difference which option is chosen.

The baseline prediction for AUC and AUPRC is .5. The reason is because assuming a dummy classifier classifies a dataset with a random guess, each prediction should tends towards the diagonal line in the ROC curve (which represents the line between a good / bad classification). For similar reasons, the baseline for AUPRC would be .5 because random predictions would create a straight horizontal line equal to $\frac{P}{N}$, where P is the number of positive examples, and N is the number of negative examples.

Problem 7.

Strain works best with PCA since its max area values are AUC = .89, AUPRC = .68

Medium works best with feature selection with AUC = .79, AUPRC = .13

Environmental pertubation works best with t-SNE with AUC = .86, AUPRC = .54. Note that PCA does have a similar but slightly lower area values.

Genetic pertubation works best t-SNE with AUC = .95, AUPRC = .81