

## Regarding model selection and L1 vs. L2 vs. elastic net vs. non-regularized regression

Hello class,

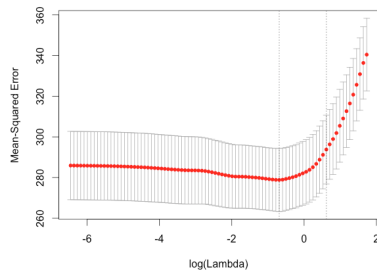
There have been a lot of questions regarding model selection, and how to decide whether:

1. regularized regression performs better than non-regularized regression;
2. L1 regularization or L2 regularization is preferable.
3. \*\* See followup for  $R^2$  discussion.

We can use the mean squared error to compare between models, but since our goal is to increase the predictive power of the model, we might want to split the data into a training set and a test set in order to get a true measure of the "goodness" of our model on unseen data.

We perform a train-test split, then fit the three models - ordinary least squares (non-regularized), lasso regression (L1-regularized), and ridge regression (L2-regularized), using our training data  $X_{train}$ , and predicting values  $\hat{y}_{train}$  of the dependent variable.

We do model selection to determine the best value of the regularization constant  $\lambda$  in each case via  $k$ -fold cross-validation on  $X_{train}$ , predicting values  $\hat{y}_{train} = X_{train} \hat{\beta}$  and recording the mean squared error (MSE), which we plot for each value of  $\log(\lambda)$  with its average value (shown as a red marker) and standard deviation over all folds (shown as error bars), similarly to this figure:



We then select a reasonable value in the range  $[\lambda_{min}, \lambda_{1se}]$ , where  $\lambda_{min}$  is the value at which the MSE is minimized, and  $\lambda_{1se}$  is the value at which the MSE is within 1 standard deviation of the minimal MSE. The first vertical line in the plot is at  $\log(\lambda_{min})$ , and the second one at  $\log(\lambda_{1se})$ .

Those values, plus the value of the MSE at those values, can be found in `cv.glmnet` using `cv.fit$lambda.min` (respectively, `cv.fit$lambda.1se`) and `min(cv.fit$cvm)` (respectively, `cv.fit$cvm[cv.fit$lambda == cv.fit$lambda.1se]`).

We then proceed to predict values of  $y_{test}$  based on  $X_{test}$  with our selected "best" value of  $\lambda$  for each model, and calculate the MSE using the predicted values on the test data,  $\hat{y}_{test}$ .

```
# Do a test-train split (e.g., 70% train, 30% test)
n = nrow(X)
split = 0.7
train = sample(1:n, round(n * split))
y = latitude
Xtrain = X[train, ]
Xtest = X[-train, ]
ytrain = y[train]
ytest = y[-train]

# Train regression models
# Find best regularization constant for each model via cross-validation (default = 10 folds)
ols = lm(ytrain ~ as.matrix(Xtrain))
lasso = cv.glmnet(as.matrix(Xtrain), ytrain, alpha = 1)
ridge = cv.glmnet(as.matrix(Xtrain), ytrain, alpha = 0)
elasticnet = cv.glmnet(as.matrix(Xtrain), ytrain, alpha = 0.5)

# Test regression models after finding best regularization constant for each
betahatols = coef(ols)
yhatols = cbind(rep(1, n - length(train)), as.matrix(Xtest)) %*% betahatols
yhatlasso = predict(lasso, as.matrix(Xtest), s = bestlambdalasso)
yhatridge = predict(ridge, as.matrix(Xtest), s = bestlambdaridge)
yhatelastic = predict(elasticnet, as.matrix(Xtest), s = bestlambdanet)

# Compare MSE on test data
sum((ytest - yhatols)^2) / nrow(Xtest)
sum((ytest - yhatlasso)^2) / nrow(Xtest)
sum((ytest - yhatridge)^2) / nrow(Xtest)
sum((ytest - yhatelastic)^2) / nrow(Xtest)
```

In general, we expect lasso regression to perform well on high-dimensional data, as it induces sparsity in the model, in that it "shrinks" to zero the coefficients of non-significant variables, and thus reduces the complexity of the model (lower number of parameters => lower complexity => lower capacity => higher generalization power).

A disadvantage of the lasso regression is that it will not necessarily yield good results in presence of high collinearity among the independent variables, as it tends to select only one variable among a group of variables with high pairwise correlations.

If several independent variables are correlated, it will tend to retain one for the model and omit the others, as it relies on the fact that their effect on the dependent variable  $y$  can be

explained through the retained variable.

However, sometimes this could lead to models that have worse predictive power than ridge regression models, so if you see that in your experiments, don't right away assume you did something wrong.

In general, ridge regression achieves a good compromise between bias and variance by: 1) retaining all variables in the model; 2) making sure the coefficients of the model do not "grow wild" to very large positive or very small negative numbers.

To see why graphically, go back to @56 (the polynomial curve fitting example). If we allow the coefficients of the regression model to be unconstrained and do not penalize their L2 norm, we are allowing for very high variance between different models built on data from the same distribution, and proneness to overfitting.

At the same time, ridge regression will not produce sparse results, meaning that lasso regression should be used for variable selection.

A compromise between the pros and cons of the two is to use elastic net regression - penalizing a regression with a mixed norm using both L1 and L2 penalty, and trading off between them via a parameter  $0 \leq \alpha \leq 1$ , which can be selected experimentally.


Lastly, keep in mind that regularization is highly-susceptible to outliers in the model, so it might be a good idea to remove them, if you want to be able to make straightforward comparisons between different models.


hw6


Updated 3 hours ago by Mariya Vasileva

### followup discussions for lingering questions and comments

☐ Resolved ☒ Unresolved


 **Anonymous** 19 hours ago  
So we must report performance of the regularizer on the test data? Can we just read the mean squared error from the plot? By the way, is it a must to remove the outliers? Since the professor says that figuring out outliers in high dimensional space is quite hard and it's not required for this assignment.

 **Mariya Vasileva** 19 hours ago No, you don't have to handle outliers, but it would help explain why some of your results may not match what you expect (e.g., if regularization is not performing well, etc). There can be different reasons, as suggested by the post, of why regularization may not produce a better model than ordinary least squares regression. But removing outliers can help identify the reason.


 **Anonymous** 19 hours ago So we must report performance of the regularizer on the test data? Can we just use the mean squared error we get directly from the cross validation?

☐ Resolved ☒ Unresolved

 **Anonymous** 19 hours ago  
is it possible that my  $R^2$  for regularized regression is smaller than my  $r^2$  for non regularized regression ?

 **Anonymous** 19 hours ago Mine is like this.

 **Anonymous** 19 hours ago Mine too.

 **Mariya Vasileva** 3 hours ago In my personal opinion,  $R^2$  is not a particularly useful quantity to calculate in evaluating how good a regularized regression fit is. The intuitive reason would be that the purpose of  $R^2$  is to measure how much of the variance in the dependent variable our model is able to explain, given by the ratio:


$$R^2 = \frac{\sum_i (\hat{y}_i - \bar{y})^2}{\sum_i (y_i - \bar{y})^2}$$


However, regularization is aimed at increasing the predictive power of the model, and something like "prediction error", i.e. MSE on test data, or cross-validated MSE, would be a more informative metric.

In particular, since the goal of L1 regularization is generally to reduce the number of parameters, and the goal of L2 regularization is to find an optimal bias-variance tradeoff, we could expect that a model built with either regularization scheme (or a mixture of both, in elastic net regression) would not be able to explain the variance in the dependent variable as well as a model that includes all variables (ordinary least squares) and does not impose restrictions on the regression coefficients.

However, we would expect the regularized model to predict on new data with lower error than ordinary least squares, as the goal of regularization (most often) is to increase the generalization power of our model.

That being said, in addressing the question "Is regularized regression better than non-regularized regression?", you have to define whether your objective is: 1) lower prediction error on new data (one way to measure this is MSE on new data); or 2) higher capacity model that produces better fit (one way to measure this is using  $R^2$ ).

 **Anonymous** 3 hours ago Is it normal that my regularized predictor with outliers unremoved has a higher MSE?

 **Mariya Vasileva** 3 hours ago Maybe those points were not outliers? Remember, it is hard to decide whether a point is truly an outlier for high-dimensional data, so we can use a variety of tools (Cook's distance, leverage, and standardized residuals vs. leverage plots are only tools that help identify influential or suspicious data points - they could be true outliers, or they could be important data points and suggest unmodelled effects).