**Regularization II: Ridge**

Lasso is great for feature selection, but when building regression models, Ridge regression should be your first choice.

Recall that lasso performs regularization by adding to the loss function a penalty term of the *absolute* value of each coefficient multiplied by some alpha. This is also known as L1L1 regularization because the regularization term is the L1L1 norm of the coefficients. This is not the only way to regularize, however.

If instead you took the sum of the *squared* values of the coefficients multiplied by some alpha - like in Ridge regression - you would be computing the L2L2 norm. In this exercise, you will practice fitting ridge regression models over a range of different alphas, and plot cross-validated R2R2 scores for each, using this function that we have defined for you, which plots the R2R2score as well as standard error for each alpha:

def display\_plot(cv\_scores, cv\_scores\_std):

fig = plt.figure()

ax = fig.add\_subplot(1,1,1)

ax.plot(alpha\_space, cv\_scores)

std\_error = cv\_scores\_std / np.sqrt(10)

ax.fill\_between(alpha\_space, cv\_scores + std\_error, cv\_scores - std\_error, alpha=0.2)

ax.set\_ylabel('CV Score +/- Std Error')

ax.set\_xlabel('Alpha')

ax.axhline(np.max(cv\_scores), linestyle='--', color='.5')

ax.set\_xlim([alpha\_space[0], alpha\_space[-1]])

ax.set\_xscale('log')

plt.show()

Don't worry about the specifics of the above function works. The motivation behind this exercise is for you to see how the R2R2 score varies with different alphas, and to understand the importance of selecting the right value for alpha. You'll learn how to tune alpha in the next chapter.