Tuning Mixtures and Penalties

The Penalty Hyperparameter

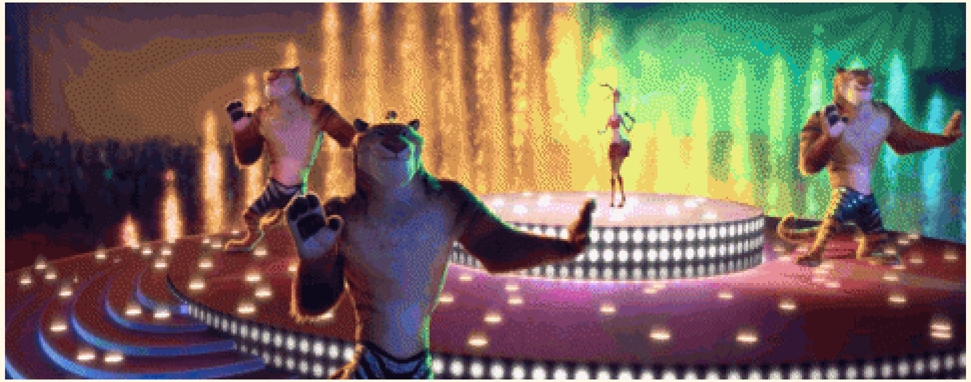
Penalty Hyperparameter

Recall: In ridge and LASSO regression, we add a *penalty* term that is balanced by a parameter λ

$$\text{minimize:} \quad RSS + \lambda * \text{penalty}$$

Penalty Hyperparameter

What is the "best" choice of λ?




Disney
ZOOTOPIA
SHAKIRA
TRY EVERYTHING

Penalty Parameter

Last class, we tried two LASSO models:

```
lasso_1 <- linear_reg(penalty = 0.1, mixture = 1) %>%
  set_engine("glmnet") %>%
  set_mode("regression")

lasso_5 <- linear_reg(penalty = 0.5, mixture = 1) %>%
  set_engine("glmnet") %>%
  set_mode("regression")
```

Penalty Parameter

We found that larger penalty parameter led to more coefficients of 0 (i.e., excluded variables):

```
## # A tibble: 63 x 3
##    term        estimate penalty
##    <chr>          <dbl>   <dbl>
##  1 (Intercept) 3.81         0.1
##  2 Creative    0.0252       0.1
##  3 Energetic   0.000327     0.1
##  4 Tingly      0            0.1
##  5 Euphoric    0.115        0.1
##  6 Relaxed     0.188        0.1
##  7 Aroused     0            0.1
##  8 Happy       0.268        0.1
##  9 Uplifted    0.108        0.1
## 10 Hungry      0            0.1
## # ... with 53 more rows
```

```
## # A tibble: 63 x 3
##    term        estimate penalty
##    <chr>          <dbl>   <dbl>
##  1 (Intercept)     4.32     0.5
##  2 Creative        0        0.5
##  3 Energetic       0        0.5
##  4 Tingly          0        0.5
##  5 Euphoric        0        0.5
##  6 Relaxed         0        0.5
##  7 Aroused         0        0.5
##  8 Happy           0        0.5
##  9 Uplifted        0        0.5
## 10 Hungry          0        0.5
## # ... with 53 more rows
```

Penalty Parameter

$\lambda = 0.1 \rightarrow$ 7 variables kept, 56 dropped

$\lambda = 0.5 \rightarrow$ 0 variables kept, 63 dropped

Prediction: What penalty leads to the best cross-validated prediction accuracy?

(`tune()` and `tune_grid()` as usual!)

Penalty Parameter

RMSE:

Penalty Parameter

One wrinkle in tuning: The automatic grid chooses values on a log scale, not evenly spaced:

```
lam_grid <- grid_regular(penalty(), levels = 1
lam_grid
```

```
## # A tibble: 10 x 1
##          penalty
##            <dbl>
##  1 0.0000000001
##  2 0.00000000129
##  3 0.0000000167
##  4 0.000000215
##  5 0.00000278
##  6 0.0000359
##  7 0.000464
##  8 0.00599
##  9 0.0774
## 10 1
```

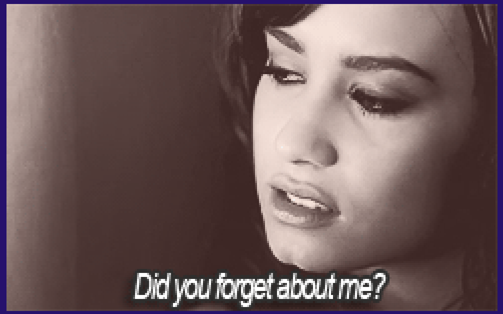
```
lam_grid_2 <- grid_regular(penalty(c(0, 0.5),
                           levels = 10)
lam_grid_2
```

```
## # A tibble: 10 x 1
##    penalty
##      <dbl>
##  1  0
##  2  0.0556
##  3  0.111
##  4  0.167
##  5  0.222
##  6  0.278
##  7  0.333
##  8  0.389
##  9  0.444
## 10  0.5
```

Penalty Parameter

RMSE, smaller grid:

Don't forget about Interpretability!



How many variables do we want to retain?

Number of Variables kept

Penalty Parameter

Because our regularized methods are de-prioritizing RMSE, they will rarely give us better prediction residuals.

So, why do it?

LASSO -> Variable selection

If we can achieve *nearly* the same predictive power with *many* fewer variables, we have a more interpretable model.

Try it!

Open Activity-Variable-Selection from last class

Tweak your choice of penalty in your LASSO regression until you get approximately the same number of variables as you did via stepwise selection.

Are they the same variables?

Model stability

Model stability

Consider dividing the dataset into 3 randomly split subsets.

We fit a linear model on *all* predictors for each subset.

Should we expect similar answers?

What's happening?

When we have **many variables**, there is probably some *collinearity*

Combinations of variables contain *the same information*.

Which ones should we use?

The model is very *unstable* with the particular sample.

Ridge Regression

Ridge regression

Why does the ridge penalty help?

It reduces the variance of the coefficient estimates.

It lets all the variables "share the load" instead of putting too much weight on any one coefficient.


The bill comes due. Always.

Ridge regression

There is no free lunch!

Lowering the variance of the estimates increases the bias.

In other words - we aren't prioritizing prediction or RMSE anymore. Our y-hats are not as close to our y's.

Elastic Net

Elastic Net

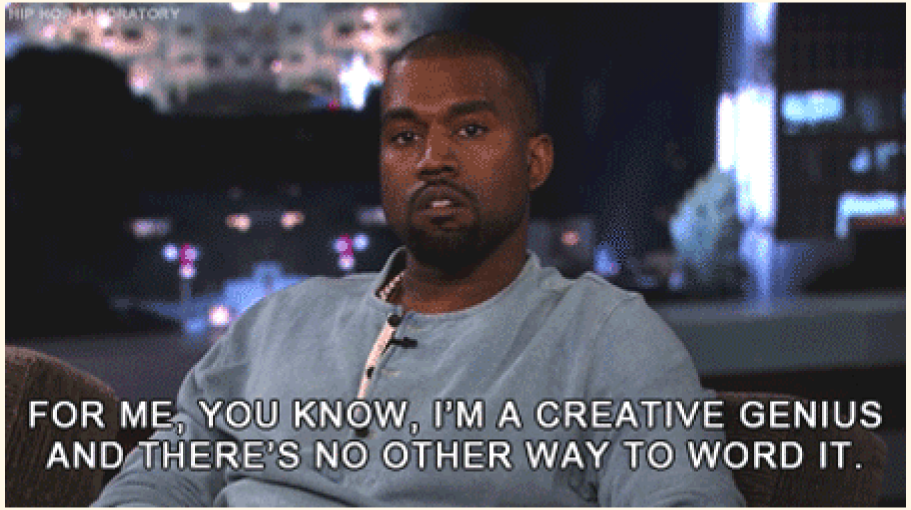
What if we want to reduce the number of variables AND reduce the coefficient variance??

Elastic Net

We'll just use two penalty terms:

$$\text{RSS} + (\text{lambda}/2)(\text{LASSO penalty}) + (\text{lambda}/2)(\text{Ridge penalty})$$

When we do half-and-half, this is called "Elastic Net".


FOR ME, YOU KNOW, I'M A CREATIVE GENIUS
AND THERE'S NO OTHER WAY TO WORD IT.

Mixtures of penalties

Why half-and-half? Why not 1/3 and 2/3? 1/4 and 3/4???

$$RSS + (\alpha)(\lambda)(\text{LASSO penalty}) + (1 - \alpha)(\lambda)(\text{Ridge penalty})$$

α is the `mixture` parameter.

Try it!

Open Activity-Variable-Selection from last class

Tune both the mixture and the penalty parameters.

Plot the RMSE and/or R-squared across a few penalties (at one mixture) and across a few mixtures (at one penalty)

Try it!

Recall: We wanted to predict the Type of cannabis from the descriptor words.

Consider only Indica vs. Sativa (no Hybrid)

Can you combine logistic regression with LASSO to tell me which words best separate Indica and Sativa?

How does this compare to what you find with a decision tree?