Ridge Logistic Regression for Preventing Overfitting
Images are made up of pixels – tiny dots with constant colour.

A grayscale image is actually can be represented as an array of numbers between 0 and 1 (0 for black, 1 for white, numbers between 0 and 1 for different shades of gray.)
Image Classification

• Suppose we have images of 2 different people
• For a new image, want to know which of the 2 people it is
• Covariates: a vector of all the brightnesses of the grayscale image

\[
\begin{align*}
\begin{array}{ccccccccc}
  x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\
  x_{12} & x_{13} & x_{14} & x_{15} & x_{16} & x_{17} & x_{18} & x_{19} \\
  x_{24} & x_{25} & x_{26} & x_{27} & x_{28} & x_{29} & x_{30} & x_{31} \\
  x_{36} & x_{37} & x_{38} & x_{39} & x_{40} & x_{41} & x_{42} & x_{43} \\
  x_{48} & x_{49} & x_{50} & x_{51} & x_{52} & x_{53} & x_{54} & x_{55} \\
  x_{60} & x_{61} & x_{62} & x_{63} & x_{64} & x_{65} & x_{66} & x_{67} \\
  x_{72} & x_{73} & x_{74} & x_{75} & x_{76} & x_{77} & x_{78} & x_{79} \\
  x_{84} & x_{85} & x_{86} & x_{87} & x_{88} & x_{89} & x_{90} & x_{91}
\end{array}
\end{align*}
\]

• $X = \{x_0, x_1, x_2, \ldots, x_{91}\}$

$Y_i$: 1 if it’s person A, 0 if it’s person B
Logistic Regression

- $Y_i \sim Binomial(X_i \beta)$
- Find $\beta$ such that the Log Likelihood is maximized
  - $\log P(y|\beta, x) = \sum_{i=1}^{m} y_i \log \left( \frac{1}{1+\exp(-x_i \beta)} \right) + (1 - y_i) \log \left( \frac{\exp(-x_i \beta)}{1+\exp(-x_i \beta)} \right)$
Classification

- Person A, if $X_i \beta > 0$ and Person B otherwise
What do the $\beta$s look like?

• Remember $X =$

• Now go back and construct an image from the $\beta$s
Overfitting

• In a small dataset, maybe a pixel in the corner \((x_1)\) of pictures of person A is always be smaller than 0.1, and for person B is always larger than 0.9
  • This would tend to make \(\beta_1\) very large and negative
    • Recall perfect separation
  • Would hurt classification performance on new data
Ridge Logistic Regression

• Minimize $NLL + \frac{\lambda}{2} \sum_{i=1}^{K} \beta_i^2$
  • (NLL = Negative Log-Likelihood)
• $\lambda = 0$ is what we did before
• $\lambda > 0$ means that we are *not* minimizing the NLL. Instead, we are trying to make the NLL as small as possible, while still making sure that the $\beta$s are not too large
  • Tradeoff between good fit (large log-likelihood) and good generalization (good performance on new data)
    • If we expected the “correct” $\beta$s to not be very large, makes sense to force them to be small
Ridge Logistic Regression

• Select $\lambda$ using cross-validation (usually 2-fold cross-validation)
  • Fit the model using the training set data using different $\lambda$’s. Use performance on the validation set as the estimate on how well you do on new data. Select the $\lambda$ with the best performance on the validation set.
Ridge Logistic Regression and Inference

Is the pixel at location (20, 30) in images of John Lennon usually darker than the one in images of Paul McCartney?

• Look at the $\beta$ that corresponds to the pixel at (20, 30)
• If we are using ridge regression, cannot obtain the standard error in the usual way
• If we really believe that the $\beta$ cannot be too large, that should estimate both the standard error and the point estimate of $\beta$ – people generally use the Bayesian Inference framework when using Ridge Regression