Logistic Regression

To get some practice with logistic regression, we'll first be looking at American election data from the National Election Study from 1948 through 2002. Afterward, we'll be taking a look at a classic classification task: email spam filtering.

A note on `caret`

In the following, use the `caret` package to obtain cross-validated estimates of α and λ for regularized logistic regression. You can simply do 5-fold cross-validation without any repeats (instead of 10-fold with 3 repeats). (If you want better predictive power, then do a *second* run of `train()` where you look more finely at values of (α, λ) close to the optimal values you found with a coarser search. However, it's not that important.)

When calling `train()`, you should set `classProbs=TRUE` in the control parameters. Also, set `summaryFunction` to either `twoClassSummary` or `multiClassSummary` depending on whether you want your metric of model quality to be area under the ROC or log loss.

National Election Study Analysis

If you get stuck, look at section 5.1 of Gelman and Hill as necessary.

For the tasks described below:

- You should use regularized logistic regression in conjunction with the `caret` package.
- Consider adding interaction terms to your fits; if you do so, again use the `caret` package to determine whether or not adding them improves your model.
- Plot the ROC curves for your best models.

- Finally, interpret the coefficients of your models.

Feel free to play around with the data and find interesting relationships.

Getting started

Refer back to the assignment on factors if you need a refresher on how factors work.

- Use the `read.dta()` function from the `foreign` library to load `elections.dta`.
- Select the columns `year`, `age` through `religion`, `vote`, and `presvote`. Restrict to years with a presidential election.
- If you check the levels of the dataframe's factors (with `lapply(df[sapply(df, is.factor)], levels)`), you'll see that the data needs some cleaning. Modify the `levels()` of each factor so the descriptive text is more concise. You can replace missing values with `NA`s (using `addNA()`), but note that a missing survey response often carries its own information and corresponds to its own category, and as such should not by default be replaced with a `NA` and subsequently imputed.
- Replace `NA`s with column means if the column they're in is numeric. If the column is a factor, instead replace each `NA` with a randomly chosen level of the factor such that the proportion of each factor level to the number of entries stays unchanged after replacement.
- You'll want to expand each factor out into *indicator variables*. You can use the function you previously wrote in the assignment on factors to do so. Alternatively, you can learn how to use the `dummies` package or how to use `model.matrix` to do so.

Exploring the data

Before you do any data analysis, it's typically a good idea to do some basic visualizations and get a sense of what you're working with.

Use R's `mosaicplot()` function to make a couple mosaic plots from the cleaned and simplified dataset. For example, try:

```
mosaicplot(table(df$income, df$presvote))
```

Things to predict

- Predict support for George H. W. Bush in the 1992 election. (Restrict consideration to people who actually voted!)

- Predict party support for different years and look at how the coefficients of the predictors change over time.
- For voters who didn't vote, predict how they *would have* voted. If you aggregate these predictions by election year, how do they appear to change over time?

The CSDMC2010 SPAM Corpus

You'll be looking at the data from the CSDMC2010 SPAM Corpus. The data is in `spam-emails.csv`, with `spam-emails-key.txt` giving the correct classification as spam or not-spam.

- Use the `tm` package to construct a `DocumentTermMatrix` from the data, where each row represents a document, each column represents a word, and each entry contains the word frequency for the associated word in the associated document.
- Use the `caret` package to find the optimal values of (λ, α) for regularized logistic regression in the classification of the documents.
- Using the optimal hyperparameters you found, train a regularized logistic regression model on the whole dataset. Look at the ROC curve.

Some things to keep in mind (read these before you begin):

- For a reference about text preprocessing with `tm`, look here or here.
- Not every single email in the dataset successfully made it into `spam-emails.csv`. You'll want to do an `inner_join()` (from `dplyr`) on the dataset and the classification key, retaining only the rows which are successfully matched.
- Remove columns corresponding to words that show up fewer than 10 times in total throughout the entire corpus.
- The matrix of documents and word frequencies is very sparse – don't scale it! If you do so, it will make it non-sparse, which is bad!