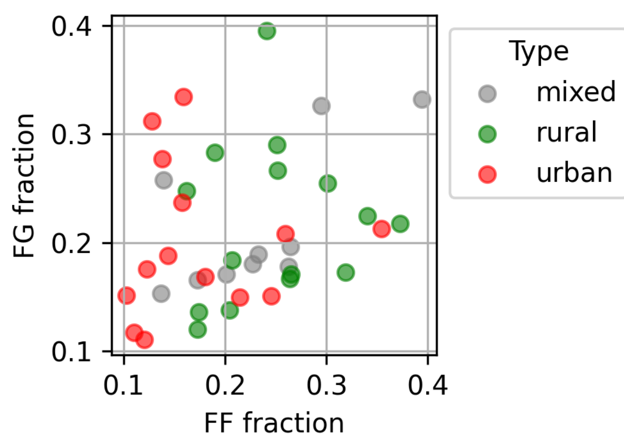
A linear classifier.

This section won't actually do anything that is different from the logistic regression we described in the previous section; instead it presents the same mathematics from a slightly different perspective. In the logistic regression example we started with a one-dimensional example, here we will look at another example, but this time a higher dimensional one, based on data from Irish elections.

As described previously, elections in Ireland use an elegant voting system called Single Transferable Vote that allows for constituency-based proportional voting. There are multi-seat constituencies and votes transfer as candidates are elected or eliminated, so it is a simplification to reduce the votes to the proportion for each party. However, for our purposes here this is what we have done. Based on the 2020 election[1] this figure show the fraction of 'first preference' vote for two of the main parties, Fianna Fáil and Fine Gael, for all 39 constituencies.



The constituencies have been labelled 'urban', 'mixed' and 'rural' and this has been marked by colour. Don't take these labels too seriously; I guess it based on my own knowledge.

It is clear there is a relationship between the voting pattern and settlement type with rural area more likely to vote for these two parties. There are exceptions, in voting local issues often change the result; the rural con-

[1] data.gov.ie/dataset/candidate-details-for-general-election-2020

stituency Galway-Roscommon, for example, has three TDs[2], one Sinn Féin, one who was in Fine Gael who left in protest following the closure of a local hospital and another who is an activist opposed to restrictions on the cutting of turf, a significant issue in an area with large amounts of bog. Similarly, the urban constituency with the largest Fianna Fáil vote is Cork South-Central; this is the constituency of Micheál Martin, who is leader of Fianna Fáil, in the event he went on to become Taoiseach[3] from 2020 to 2022.

Lets just concentrate on the urban and rural consistituencies and imagine we want to guess the settlement type of an unknown constituency based on the Fiánna Fail and Fine Gael votes. The obvious approach to that is to mark areas on the vote-fraction plane corresponding to rural and urban constituencies and, in a linear model that means dividing the plane using a straight line. In fact, this is the same as the logistic regression tasks. In logistic regression we modelled the probability using a logistic function:

$$p(\text{rural}) = \sigma(\beta_0 + \beta_1 x_1 + \beta_2 x_2) \tag{1}$$

where, in our current context, x_1 is the fraction voting Fianna Fáil and x_2 the fraction voting Fine Gael.

Next we need a loss function; as in the other examples we have looked at there are different ideas of what the loss function should be. A common example is the log-likelihood. To describe this we will refer to the probability as the *model*; in this terminology the model is a map from a data point $\mathbf{x}^a$, where I am using a as a trial index to label different data points,[4] to a probability. For a linear classifier the model is parameterized by the linear coefficients β_0, β_1 and so on. In our case

$$\mathbf{x}^a \mapsto p(\mathbf{x}^a|\beta_0, \beta_1, \beta_2) \tag{2}$$

In a full Bayesian treatment, we might ask how probable the model is given the data: $p(\beta_0, \beta_1, \beta_2|\mathbf{x}^2)$, but this can be difficult to calculate, it involves

[2]Teachta Dálá, the title given to a representative in Ireland

[3]the equivalent of Premier or Prime Minister

[4]Of course in computer science we would use this index to make a matrix x_{ia} but I am falling into the mathematicians habit of using different sorts of indices when they mean different sorts of things, a trial index is different from the index for the components of $\mathbf{x}$, mathematicians like to come up with notations that express differences like that, computer scientists, in contrast want to translate everything in code and code doesn't care about the background for an index, an index is just a way of accessing different compenents of an array.

Bayesian inversion which might require sums or integrals which often aren't tractible. Instead we use the likelihood. The likelihood is the probability of the data for the model; this is sort of back-to-front, we can ask "how likely is the data given the model'.

In the obvious notation and using our case as template example

$$L = \prod_{a \in \text{rural}} p(\mathbf{x}^a) \prod_{a \in \text{urban}} [1 - p(\mathbf{x}^a)] \tag{3}$$

where I have writen $p(\mathbf{x})$ for $p(\mathbf{x}|\beta_0, \beta_1, \beta_2)$ and using the fact that in the obvious way, if the model says the probability of rural if p then it is giving the probability of urban as $1-p$. Finally this involves lots of multiplications, this is bad for lots of reasons, but in this context the most important is that the probabilities will all be less then one and so if there is a lot of data L will be tiny, a bad property for something being stored on a computer with finite accuracy. However, ever since John Napier[5] published *Mirifici Logarithmorum Canonis Descriptio* in 1614 we've known how to get around this, we take the log, so the log-likelihood is

$$\log L = \sum_{a \in \text{rural}} \log p(\mathbf{x}^a) + \sum_{a \in \text{urban}} \log\,[1 - p(\mathbf{x}^a]) \tag{4}$$

This is big for a good choice of βs, so we use minus it, the negative log-likelihood as an objective function.

We found values for the βs by optimising this loss function. With these parameters we can then estimate a probability that any point is 'rural'; for a binary classified we can dump this nicety and just predict that a constituency is rural if $p(\text{rural}) \geq 0.5$. In fact, $\sigma(0) = 0.5$ so the decision boundary where $\hat{p}(\text{rural}) = 0.5$ corresponds to

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0 \tag{5}$$

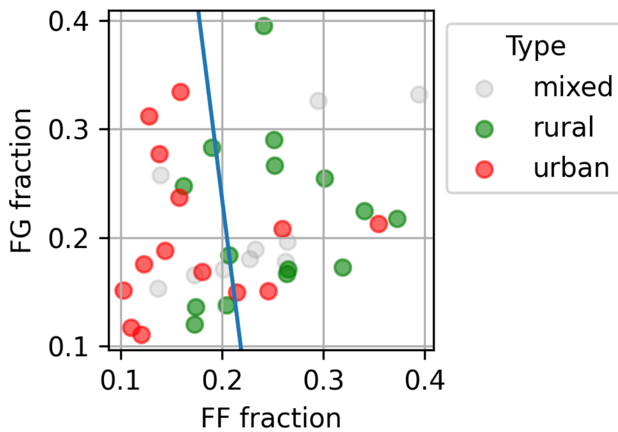
or, put another way, the line

$$x_2 = -\frac{\beta_1}{\beta_2} x_1 - \frac{\beta_0}{\beta_2} \tag{6}$$

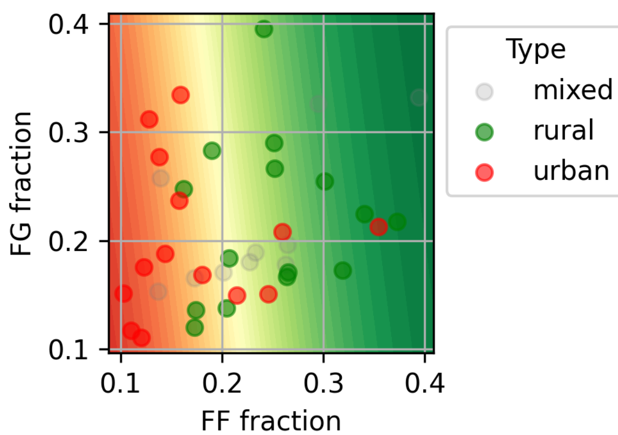
[5]John Napier invented logs as a way to speed up calculations, he saw the time that people spent multiplying as a modal challenge, it took time from them they could be using to think about God. He also invented an early version of the slide-rule which was called Napier's bones and calculated the date of the end of the world, between 1688 and 1700 he was wrong about that, which is good, I guess.

Points to the above and to the right of this line have a greater than 0.5 estimated chance of being rural, those below and to the right, less than 0.5.

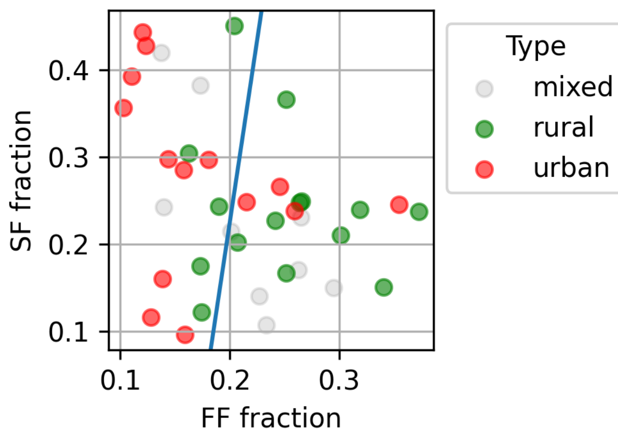
We can plot that line



We can see it has done ok, it isn't perfect, but then no line is going to be. We can look at the assigned probability using a heatmap



Clearly, it is has fitted the points as best as it can using a linear classifier and this gives a useful model of settlement type, but a linear model, perhaps unsurprisingly, has failed to account for Cork South-Central, assigning it a high probability of being rural. We can also see that the line is nearly vertical; the Fine Gael vote does not seem to be playing a significant role in distinguishing rural and urban. There is a complex history in this, Fianna Fáil is a broad party which lost a lot of support in the tumult surrounding the Great Recession it lost support, particularly to Sinn Féin in urban areas; however, Sinn Féin also has rural support so including it isn't much better



There are lots of other parties we could consider, the left-wing and social democratic parties like The Labour Party, The Green Party, People Before Profits and The Social Democrats tend to do better in cities. However, I don't want to get too distracted so lets stop. The data are on the github though if you want to look, for example, if you want to do a classifier in higher dimensions where the line will be replaced by a plane or hyperplane.

Another topic we won't consider here is the objective function, we took the log-likelihood loss from logistic regression, but there are others used for this problem. These approaches, some of which also apply to regression, aim to *regularize*, that is, come up with a fit that is less influenced by outliers and more likely to work on unseen data; these leads to topics like *ridge regression*, *lasso regularizers* and *support vector machines*. Regularization is, in general, something we will consider more in the future. In the next section though, we will look at the perceptron, which at first seems like a weird way to reformulate linear classification but turns out to be useful.

Summary

This is a straight-forward section in that it doesn't really introduce any new mathematics, all it does is replace the logistic regressor

$$\hat{p} = \sigma(\beta_0 + \boldsymbol{\beta} \cdot \mathbf{x}) \tag{7}$$

with the classification boundary corresponding to $\hat{p} = 0$:

$$\beta_0 + \boldsymbol{\beta} \cdot \mathbf{x} = 0 \tag{8}$$