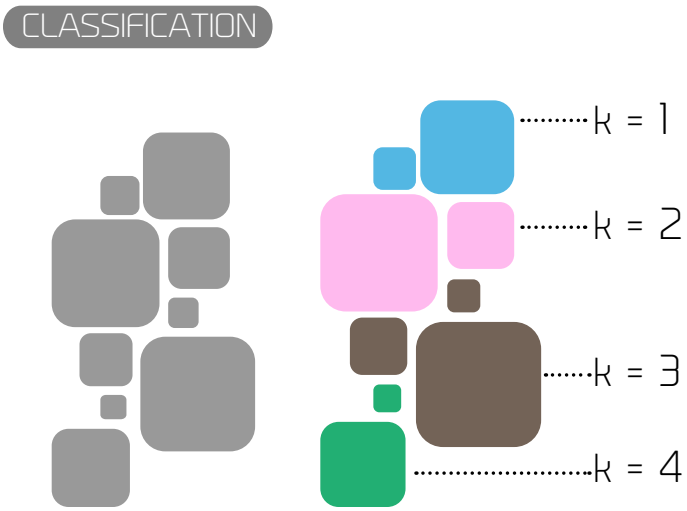
Workshop | Logistic Regression

Last week, we explored a class of models that had simple analytical and computational properties. Today we will extend that discussion to classification problems. The goal of classification is to take an input data vector and assign it to one of K classes, C_k , where k = 1,…,K.



Linear Classification |

We can of course use linear regression tools for classification, where the target, t, we wish to predict is simply a linear combination of our data and parameters,

$$t = y(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$$

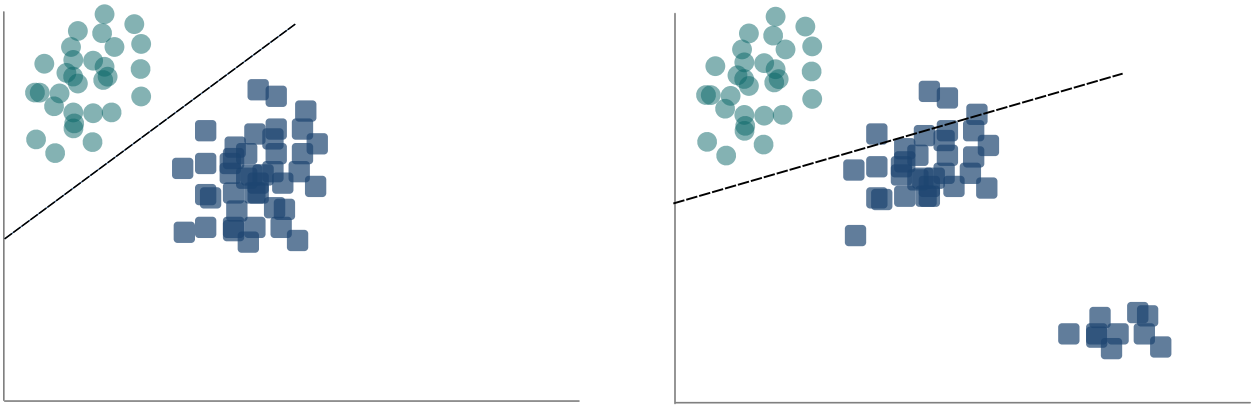
We fit this to an error function, E(w), to estimate the parameters, w. For simplicity, the most popular error function is least squares,

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2$$

There are two big issues with linear classification:

1. The number of parameters explodes in higher dimensions
2. Too sensitive to outliers.

The first of the two problems can be seen from the least squares equations. The more dimensions you have, the larger N is. The figure below illustrates the second problem,



Notice, that when data points are “too well” classified, the decision boundary is heavily skewed. We can avoid this problem with logistic regression.

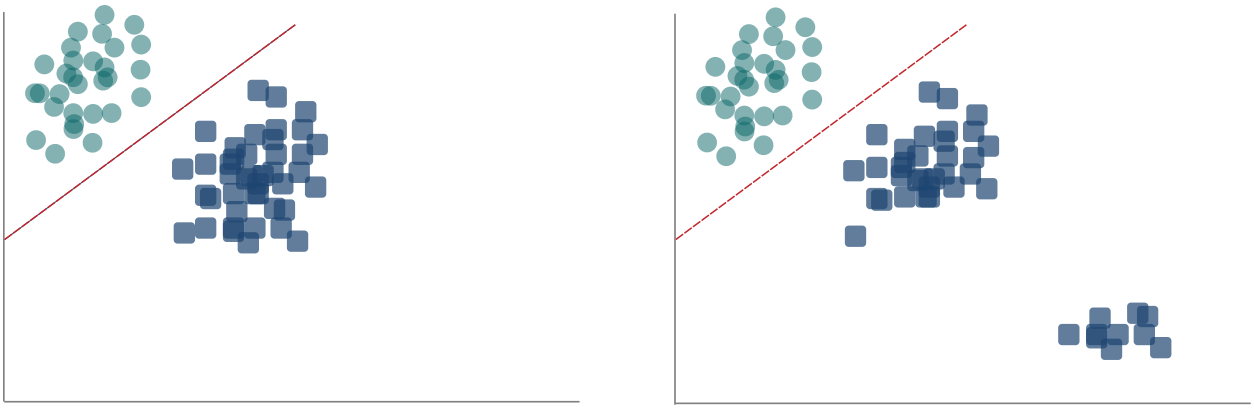
Logistic Regression I

Skipping over the proofs and derivations, the logistic sigmoid is defined as:

$$\sigma(a) = \frac{1}{1 + exp(-a)} \quad \text{where,} \quad a = \ln \frac{p(\mathbf{x}|C_1)p(C_1)}{\sum_n^N p(\mathbf{x}|C_n)p(C_n)}$$

This may not look like much, but two things come out of this:

1. There is a linear dependence in the number of parameters, meaning much faster computation
2. We’ve avoided the problem of over fitting.



Final Thoughts |

Although this is a very good solution to classification there are several problems with logistic regression

- can exhibit severe overfitting
- singularity in the sigmoid function leading to wrong estimates
- no way of favouring one solution over another