Machine learning

Classification, part 1

Joshua Loftus

Classification

- **Supervised learning** with categorical/qualitative outcomes

(in contrast to regression, with numeric outcomes)

- Often called "labels", K = number of unique classes

Label names not mathematically important - e.g. use $1, \ldots, K$

- Binary: positive/negative or 0/1 or yes/no or success/fail etc
- Binary thinking easier $\rightarrow$ bin / discretise other outcomes and do binary classification instead of regression or $K > 2$ classification (warning: information loss)
- Plots with 2 predictors, use color/point shape for outcomes

Interpretable classification

Logistic regression

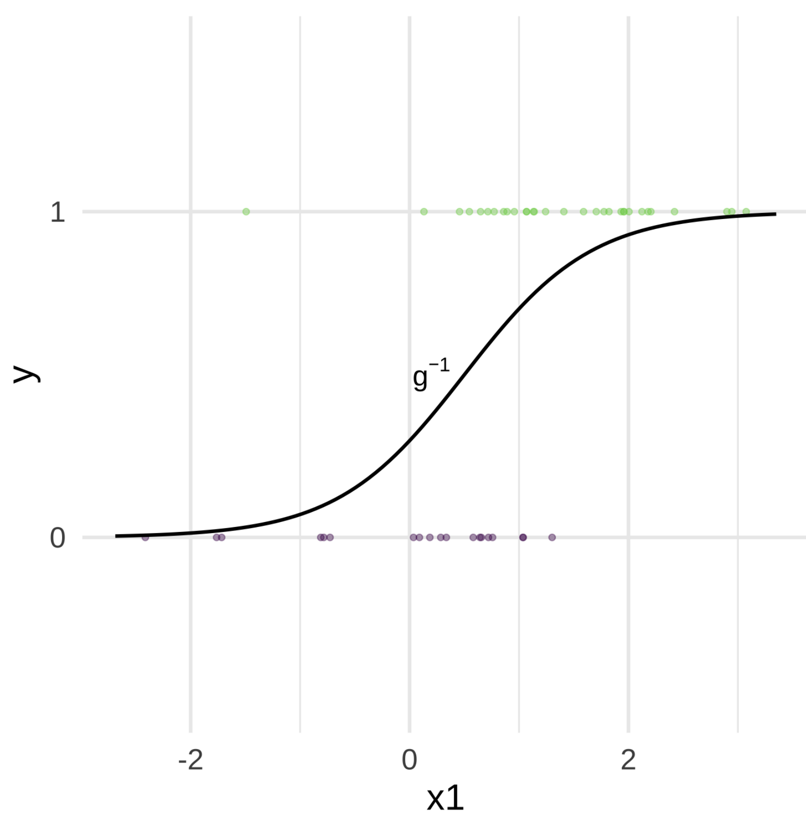
$$\mathbb{E}(Y|\mathbf{X} = \mathbf{x}) = g^{-1}(\mathbf{x}^T\beta)$$

for

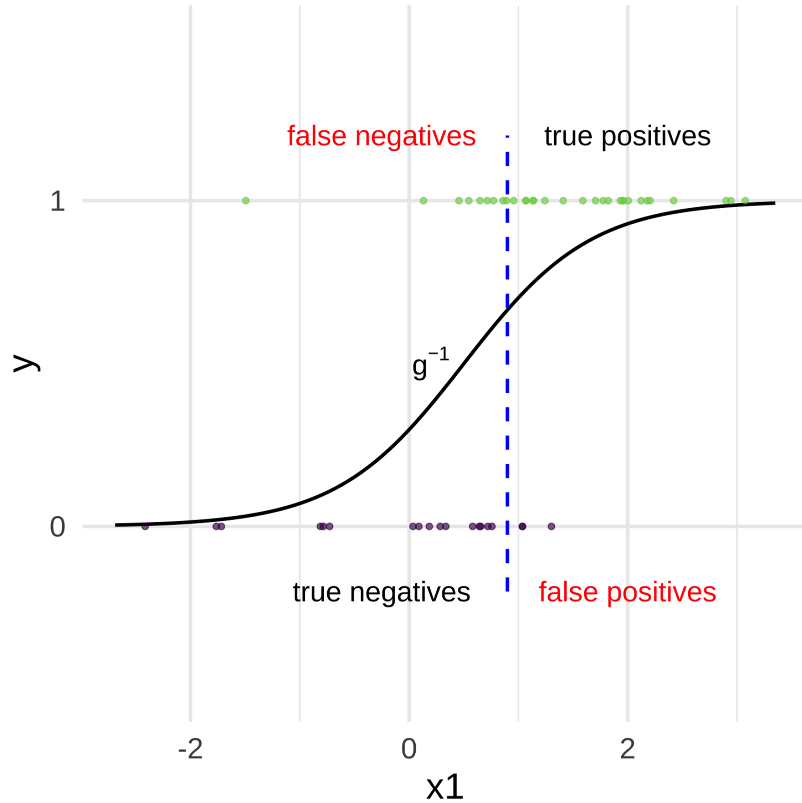
$$g(p) = \log\left(\frac{p}{1-p}\right)$$

- Other GLMs have different "link" functions g
- (Linear regression is a special case with $g = \mathrm{id}$)
- Multi-class / multinomial / "softmax" regression
- Estimation/optimization: maximum-likelihood via Newton-Raphson / IRLS

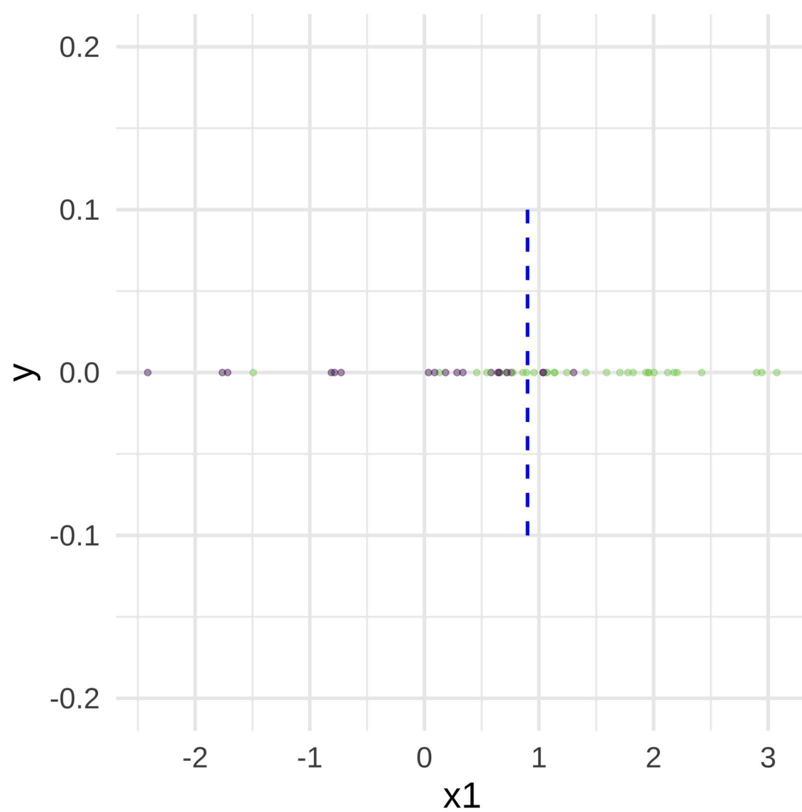
One predictor, "S curve"



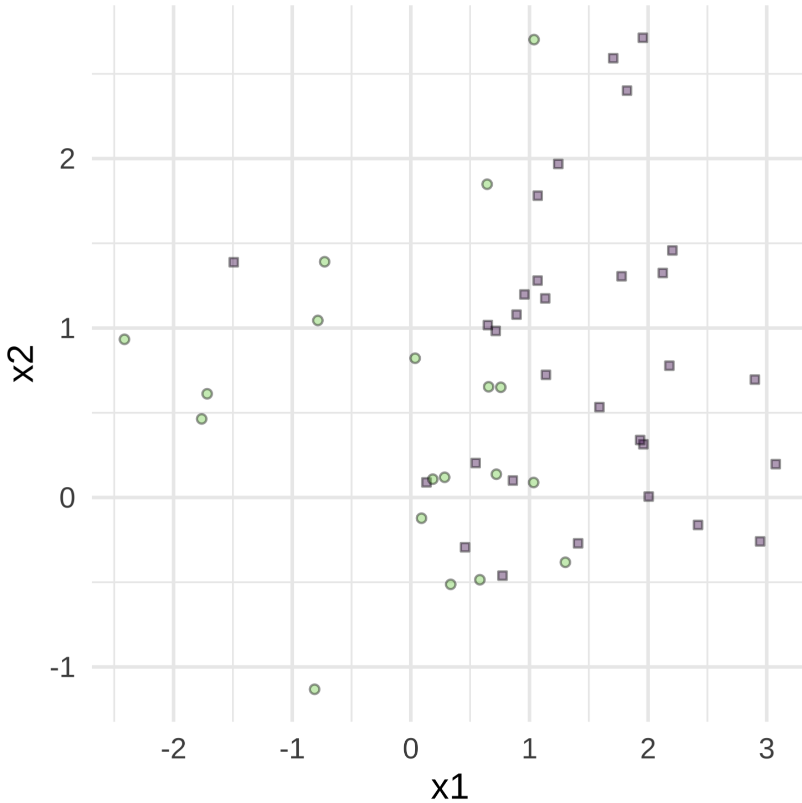
Classifications/decisions: threshold probability



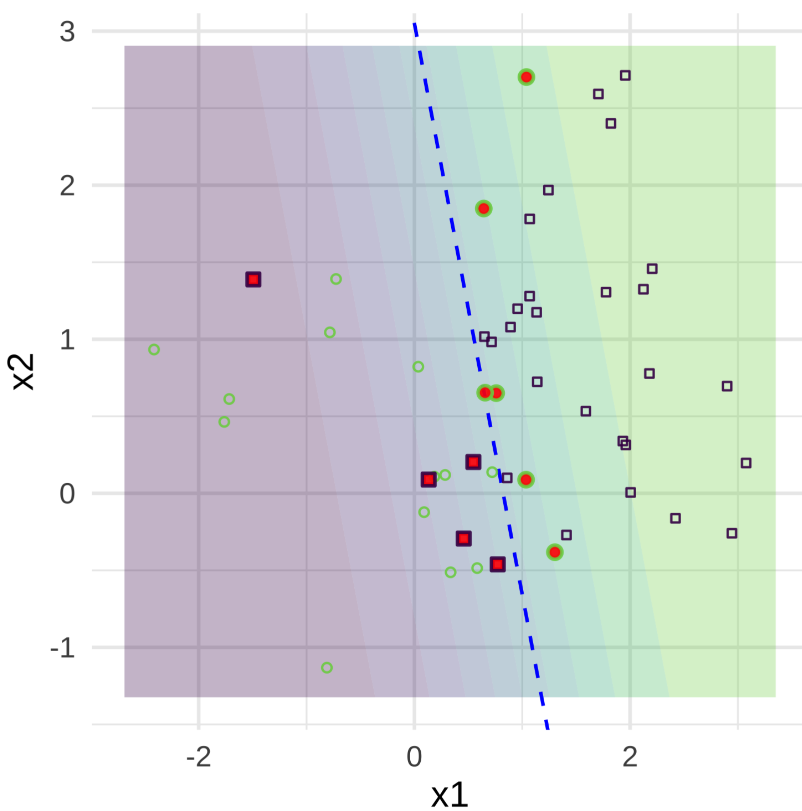
Without giving y a spatial dimension



Two predictors, binary outcome



Contours of GLM-predicted class probabilities



Classification boundaries with

$p = 3$ predictors

Boundary = plane

$p > 3$ predictors

Boundary = hyperplane

(In practice, "high-dimensional" = can't easily plot it)

Fitting/estimation

How do we estimate β? **Maximum likelihood**:

$$\text{maximize } L(\beta; \mathbf{y}|\mathbf{X}) = \prod_{i=1}^{n} L(\beta; y_i|\mathbf{x}_i)$$

(assuming the data is i.i.d.)

It's good to have some understanding of what's involved (it's not magic)

Next slide: consider a one-parameter case, one predictor and no intercept, so the calculus simplifies

Logistic regression fitting: MLE 😱 jk 😎

$$L(\beta; \mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} \left(\frac{1}{1+e^{-x_i\beta}}\right)^{y_i} \left(1 - \frac{1}{1+e^{-x_i\beta}}\right)^{1-y_i}$$

$$\ell(\beta; \mathbf{y}|\mathbf{x}) = \sum_{i=1}^{n} y_i \log\left(\frac{1}{1+e^{-x_i\beta}}\right) + (1-y_i)\log\left(1 - \frac{1}{1+e^{-x_i\beta}}\right)$$

$$\frac{\partial}{\partial\beta}\ell(\beta; \mathbf{y}|\mathbf{x}) = \sum_{i=1}^{n} y_i \left(\frac{x_i e^{-x_i\beta}}{1+e^{-x_i\beta}}\right) + (1-y_i)\left(\frac{-x_i}{1+e^{-x_i\beta}}\right)$$

$$= \sum_{i=1}^{n} x_i \left[y_i - \left(\frac{1}{1+e^{-x_i\beta}}\right)\right] = \sum_{i=1}^{n} x_i[y_i - \hat{p}_i(\beta)]$$

Set this equal to 0 and solve for β using Newton-Raphson

Newton-Raphson

- Find the roots of a function
- Iteratively approximating the function by its tangent
- Root of the tangent line is used as starting point for next approximation
- See the animation on Wikipedia

Exercise: using result from previous slide, compute the second derivative of ℓ and derive the expressions needed to apply Newton-Raphson

Logistic regression fitting: multivariate case

Newton-IRLS (equivalent) steps:

$$\begin{aligned}
\hat{\mathbf{p}}_t &= g^{-1}(\mathbf{X}\hat{\beta}_t) && \text{update probs.} \\
\mathbf{W}_t &= \text{diag}[\hat{\mathbf{p}}_t(1-\hat{\mathbf{p}}_t)] && \text{update weights} \\
\hat{\mathbf{y}}_t &= g(\hat{\mathbf{p}}_t) + \mathbf{W}_t^{-1}(\mathbf{y}-\hat{\mathbf{p}}_t) && \text{update response}
\end{aligned}$$

and then update parameter estimate:

$$\hat{\beta}_{t+1} = \arg\min_{\beta} (\hat{\mathbf{y}}_t - \mathbf{X}\beta)^T \mathbf{W}_t (\hat{\mathbf{y}}_t - \mathbf{X}\beta)$$

Note: larger weights on observations with $\hat{p}$ closer to 1/2, i.e. the most difficult to classify (***look for variations of this theme***)

See Section 4.4.1 of ESL

Inference

- MLEs $\rightarrow$ asymptotic normality for intervals/tests

`summary()`, `coef()`, `confint()`, `anova()`, etc in R

- "Deviance" instead of RSS
- Because y is 0 or 1, residual plots will show patterns, not as easy to interpret geometrically

Reference: CASI Chapter 4 for MLE theory, Chapter 8 for logistic regression and GLMs

Challenges

Separable case (guaranteed if $p > n$)

If classes can be perfectly separated, the MLE is undefined, fitting algorithm diverges as $\hat{\beta}$ coordinates $\to \pm\infty$

Awkwardly, classification is *too easy*(!?) for this probabilistic approach

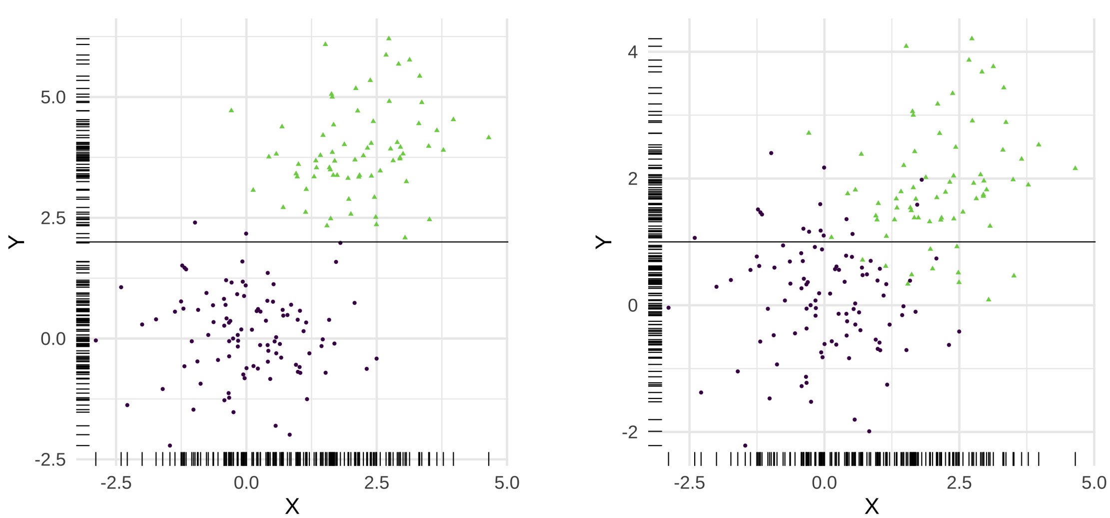
Curse of dimensionality

Biased MLE and wrong variance/asymp. dist. if $n/p \to \text{const}$, even if > 1

See Sur and Candès, (2019)

Recap: numeric outcome $\to$ categorical

Warning: *arbitrary* binning may be unwise. "*Carve nature at its joints*"



Recap

- Numeric prediction $\to$ classification

$$\hat{y} = \mathbb{I}(\hat{p} > c) = \begin{cases} 0 & \text{if } \hat{y} \leq c \\ 1 & \text{if } \hat{y} > c \end{cases}$$

- Logistic regression

Log-odds of $\hat{p}$ = linear function of x, so $\hat{p} > c \leftrightarrow x^T\beta > c'$

Linear classification boundary (hyperplanes)

- Optimization problems

Iterative algorithms (e.g. Newton-Raphon)

Adapting to "most interesting" (difficult to classify) data