Statistical Methods

Log-Linear Models, Hilary Term, 2016

Robin Evans

(based on slides by Marco Scutari)

evans@stats.ox.ac.uk
Department of Statistics
University of Oxford

April 27, 2016

Course Information

Lectures

Weeks 1 and 2: Monday and Wednesday 11am

Practical

Week 2: Friday (not assessed)

Reference Books (further references in the next slide)

AA Agresti A (2013). Categorical Data Analysis. Wiley, 3rd edition.

MN McCullagh P, Nelder AJ (1989). Generalized Linear Models. Chapman & Hall, 2nd edition.

VR Venables WN, Ripley BD (2002). Modern Applied Statistics with S. Springer, 4th edition.

Other Useful Books on Generalised Linear Models

- Christensen R (1997). Log-Linear Models and Logistic Regression. Springer, 2nd edition.
- Davison AC (2008). Statistical Models. Cambridge University Press.
- Faraway (2006). Extending the Linear Model with R. Chapman & Hall.
- Hastie T, Tibshirani R, Friedman J (2009). The Elements of Statistical Learning. Springer, 2nd edition.
- Kleinbaum DG, Klein M (2010). Logistic Regression: A Self-Learning Text. Springer, 3rd edition.
- von Eye A, Mun E-Y (2013). Log-Linear Models: Concepts, Interpretation and Application. Wiley.
- Wood SN (2006). Generalized Additive Models: An Introduction with R. Chapman & Hall.

Overview

1. Generalised Linear Models

   [MN 2; AA 4]

2. Logistic Regression

   [MN 4; AA 6; VR 7]

3. Log-Linear Regression

   [MN 6; VR 7]

4. Advanced Models

   [MN 9; AA 13]

Generalised Linear Models

Recap of Linear Models

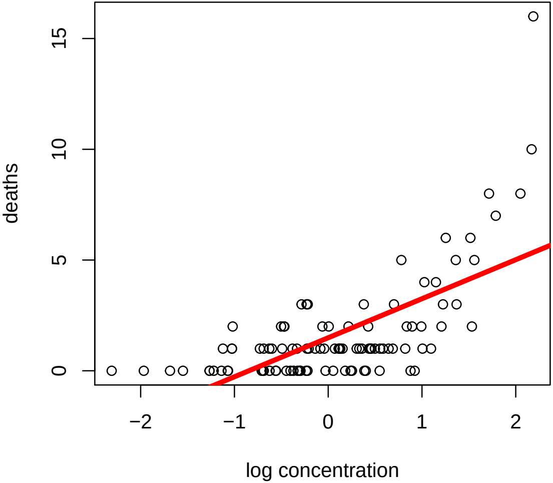
Regression is modelling how an outcome (response, dependent variable) Y_i depends on covariates (predictors, independent variables) $\boldsymbol{x}_i$. Ordinary linear model assumes:

- $\mathbb{E}Y_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p$;
- error terms $Y_i - \mathbb{E}Y_i$ are i.i.d. $N(0, \sigma^2)$.

Gaussianity assumption can be relaxed to constant variance $\sigma^2 > 0$.
This is still very restrictive: the range of responses Y_i is assumed to be unbounded, error terms are homoskedastic.
This restricts the kinds of data we can sensibly model, even with transformations. Examples:

- number of alleles of particular genotype (0, 1 or 2);
- level of particulate matter in air (positive, heteroskedastic) vs temperature;
- income (positive, skewed) vs education level;
- type of housing (categorical) vs occupation.

Linear Models aren't always so Useful



Generalised Linear Models

A more flexible class of models that tackles this problem are the generalised linear models (GLMs), made of three parts:

- an exponential family $\{f_{\theta(\mu)} : \mu \in M\}$;
- a linear predictor $\eta_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p$.
- an invertible twice differentiable link function $g : M \to \mathbb{R}$.

We then assume the response has a distribution from the exponential family (so it takes values in M) with mean $\mu_i = \mathbb{E}Y_i$ such that

$$g(\mu_i) = \eta_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p. \tag{1}$$

Ordinary linear models correspond to f_μ the Gaussian distribution with mean μ, and identity link $g(\mu) = \mu$.

Examples

Some possible choices for the response are:

- the Gaussian distribution to obtain ordinary linear models;
- the Gamma distribution for non-negative continuous responses, $Y_i \in \mathbb{R}^+$;
- the binomial distribution for binary responses, $Y_i \in \{0, 1\}$;
- the Poisson distribution for count data, $Y_i \in \mathbb{N} \cup \{0\}$.

We will concentrate on the last two, which are by far the most popular non-Gaussian GLMs.

Exponential Families

Recall that distributions in the exponential family have densities (mass functions) of the form

$$f(y;\theta) = \exp\left\{y\theta - b(\theta) + c(y)\right\} \tag{2}$$

where θ is called the canonical parameter.

We can make this slightly more flexible by adding a dispersion parameter $\psi > 0$:

$$f(y;\theta,\psi) = \exp\left\{\frac{y\theta - b(\theta)}{\psi} + c(y,\psi)\right\} \tag{3}$$

This creates an exponential dispersion family.

$b(\cdot)$ and $c(\cdot)$ are assumed known, θ, ψ unknown.

Exponential Families

Using the usual trick,

$$\begin{aligned} 0 = \frac{\partial}{\partial\theta}1 = \frac{\partial}{\partial\theta}\int f(y;\theta)dy &= \int \frac{\partial}{\partial\theta}f(y;\theta)dy \\ &= \int \psi^{-1}(y - b'(\theta))f(y;\theta)\,dy \\ &= \psi^{-1}\left(\mathbb{E}_\theta Y - b'(\theta)\right). \end{aligned}$$

Since $\psi > 0$ then $b'(\theta) = \mu(\theta) \equiv \mathbb{E}_\theta Y$.

Exercise. (Do and learn this!) Show that $b''(\theta) = \psi^{-1}\operatorname{Var}_\theta Y > 0$.
[Hint: use the same trick again.]

This shows that $b'(\theta)$ is monotonic.
Hence we can equally parameterise using θ (canonical parameterisation) or μ (mean parameterisation).

Exponential Families

Now, $b'' > 0$ shows that b is convex, and therefore $f(y;\theta)$ is a concave function of θ. This means that optimisation problems such as maximum likelihood have nice computational properties: e.g. guaranteed unique solution, can be solved with simple methods.

Note that

$$\psi^{-1} \operatorname{Var}_\theta Y = b''(\theta) = b''(b'^{-1}(\mu)) \equiv V(\mu).$$

so the variance depends upon the mean in general. This function $V(\mu)$ is called the variance function.

Example: The Normal Distribution

The most familiar example is the normal distribution, which has the form

$$\begin{aligned} f(y;\mu,\sigma^2) &= (2\pi\sigma^2)^{-1/2}\exp\left\{-\frac{1}{2\sigma^2}(y-\mu)^2\right\} \\ &= \exp\left\{\frac{y\mu-\mu^2/2}{\sigma^2}+c(y,\sigma^2)\right\}. \end{aligned}$$

where $c(y,\sigma^2) = -\frac{1}{2}\left(\sigma^{-2}y^2 + \log(2\pi\sigma^2)\right)$, and we have

$$\theta = \mu \qquad\qquad b(\theta) = \frac{\theta^2}{2} \qquad\qquad \psi = \sigma^2.$$

Note that $b'(\theta) = \theta = \mu = \mathbb{E}Y$, and $\psi b''(\theta) = \sigma^2 = \operatorname{Var} Y$ so $V(\mu) = 1$.

In this case the mean and canonical parameterisation are the same, and the variance does not vary with the mean.

Example: The Poisson Distribution

The Poisson distribution takes values in $0, 1, 2, \ldots$, and is often used to model count data (e.g. number of deaths).

$$P(Y = y) = \frac{\lambda^y e^{-\lambda}}{y!}, \qquad\qquad y \in \{0, 1, 2, \ldots\}.$$

It is parameterised by its mean λ, which is also its variance:
$\mathbb{E}Y = \operatorname{Var} Y = \lambda$.

We can write

$$f(y; \lambda) \propto \lambda^y e^{-\lambda} = \exp\left\{ y \log \lambda - \lambda \right\},$$

so the canonical parameter is $\theta \equiv \log \lambda$, and $b(\theta) = e^{\theta}$.

We also have $\mathbb{E}Y = \operatorname{Var} Y = \lambda$, so $V(\mu) = \mu$ and $\psi = 1$
(i.e. variance is same as mean but no dispersion parameter).

Example: The Binomial Distribution

For the Binomial distribution with parameters n (fixed, known) and π:

$$f(y;\pi) \propto \pi^y(1-\pi)^{n-y} = \exp\left\{y\log\frac{\pi}{1-\pi} + n\log(1-\pi)\right\}$$

so the canonical parameter is the log-odds

$$\theta = \operatorname{logit}\pi \equiv \log\frac{\pi}{1-\pi}.$$

$\operatorname{logit}(\cdot)$ is also called the logistic function.

Exercise. Check that: $b(\theta) = n\log(1+e^\theta)$, $\mu = n\pi$, $V(\mu) = n^{-1}\mu(n-\mu)$ and $\psi = 1$.

Canonical Link Functions

How should we choose the link function?

$$g(\mu_i) = \eta_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p.$$

Ideally, g maps the mean space M onto $\mathbb{R}$, so that any linear prediction is coherent. In many cases the canonical parameter space $\Theta = \mathbb{R}$, so using $g = (b')^{-1}$ is a good choice; this is the canonical link function.

This means that $\eta = g(\mu) = b'^{-1}(\mu) = \theta$, so we are modelling the canonical parameter as a linear function of the covariates.

Canonical links are a good default choice, but: g determines our mean model (i.e. how $\mathbb{E}Y$ varies with covariates), and will affect the interpretation of any regression parameters. Choose it with this in mind.

Generalised Linear Models: Poisson Response

For count data, the natural assumption is the Poisson distribution. So

$$\mathbb{E}(Y_i) = \mu_i \qquad \text{and} \qquad g(\mu_i) = \eta_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p$$

and as a link function we (ideally) want $g : \mathbb{R}^+ \to \mathbb{R}$.

Examples that are implemented in R are:

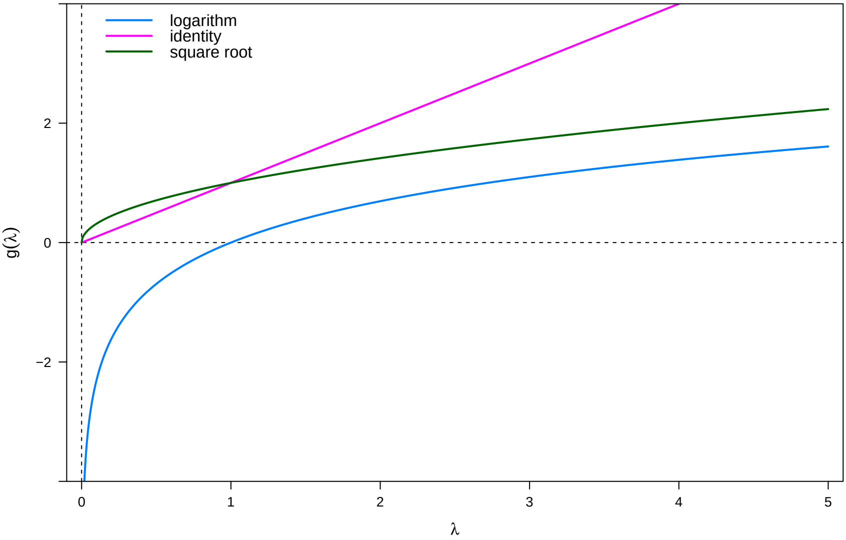
- the canonical natural logarithm

  $$g(\lambda) = \log(\lambda)$$

  (hence the name log-linear regression);
- the identity function $\qquad g(\lambda) = \lambda;$
- the square root $\qquad g(\lambda) = \sqrt{\lambda}.$

Note that the identity and square root functions will give nonsensicle parameter values if $\eta_i < 0$.

Link Functions: Logarithm, Identity and Square Root



Why These Link Functions?

The logarithm is a simple and mathematically elegant transform from $\mathbb{R}^+$ to $\mathbb{R}$, and it has an equally simple and elegant inverse in the exponential. It allows us to interpret the regression parameters in terms of multiplicative effects.

The identity is easy to interpret and suitable for "large enough" values because

$$\mathrm{Pois}(\lambda) \to N(\lambda, \lambda) \qquad \text{as } \lambda \to \infty. \tag{4}$$

In that case the responses will be very far from zero and we can structure the GLM as an ordinary linear model, without worrying about negative values.

The square root is an approximate variance-stabilising transformation (i.e. to make the variability of the values not related to their expectation, as with a normal distribution). The original is called the Anscombe transform: for $Y \sim \mathrm{Pois}(\lambda)$ and using the delta method,

$$g : y \to 2\sqrt{y + \frac{3}{8}} \qquad \text{we have} \qquad g(Y) \dot{\sim} N\left(2\sqrt{\lambda + \frac{3}{8}} - \frac{1}{4\sqrt{\lambda}}, 1\right).$$

Generalised Linear Models: Binomial Response

For a binary response we have $M = [0, 1]$ so we need $g : [0, 1] \to \mathbb{R}$. Examples:

- the canonical logistic function or log-odds

$$g(\pi) = \text{logit}(\pi) = \log \frac{\pi}{1 - \pi}; \tag{5}$$

- the probit function

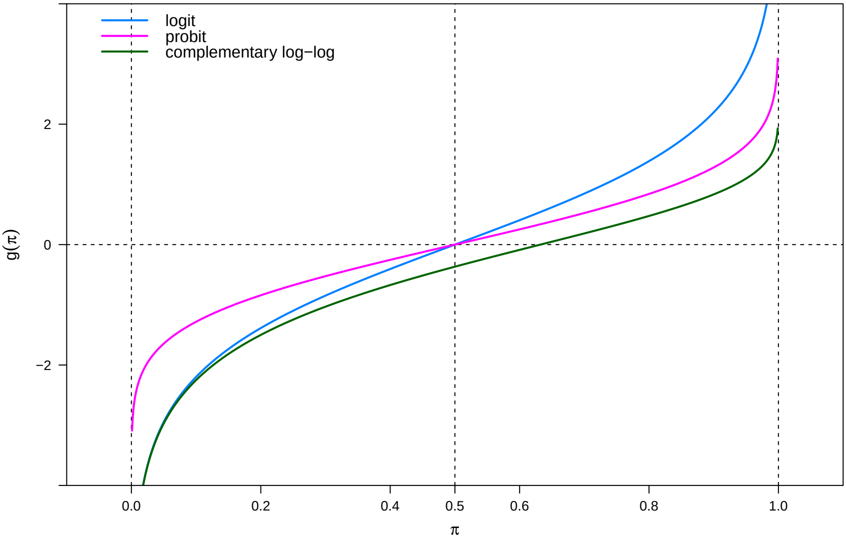
$$g(\pi) = \Phi^{-1}(\pi), \qquad \text{where } \Phi() \text{ is the Normal CDF;} \tag{6}$$

- and the complementary log-log function

$$g(\pi) = \log(-\log(1 - \pi)). \tag{7}$$

The most popular choice is (5); this is logistic regression.

Link Functions: Logit, Probit and Complementary Log-Log



Why These Link Functions?

- Logistic is convenient because it is canonical (this simplifies some calculations.)
- It is *relatively* easy to interpret using log-odds.
- Probit allows η_i to be interpreted as a z-score for $P(Y_i = 1)$.
- Complementary log-log comes from the inverse CDF of a Weibull distribution, used for modelling extreme values. It is similar to the logistic for small π, but much smaller η will give large π.

The logit function is generally preferable for convenience and interpretability. It has a closed form inverse:

$$\operatorname{logit}^{-1}(\eta) = \operatorname{expit}(\eta) \equiv \frac{e^{\eta}}{1 + e^{\eta}}. \tag{8}$$

Why Do We Prefer Canonical Link Functions?

When using a canonical link function:

- $\boldsymbol{X}^T\boldsymbol{Y}$ is the sufficient statistic for θ.
- Deriving maximum likelihood estimates is easier than with non-canonical link functions. Convexity of the optimisation problem is guaranteed, and Newton-Raphson and Fisher scoring coincide.
- Interpretation of the regression is typically intuitive (for some values of “intuitive”): think odds (for the Binomial) and multiplicative effects (for the Poisson).

The Dispersion Parameter

The variance of the data in the model is expressed as

$$\mathrm{Var}(Y) = \psi b''(\theta) \tag{9}$$

where ψ is a dispersion parameter.

It is also possible to have $\mathrm{Var}(Y_i) = \psi w_i b''(\theta_i)$ where the w_i are known weights that can be different for different observations.

NOTE: for the distributions which do not have a dispersion parameter separate from the expectation (the normal does, the binomial and the Poisson do not), fitting a generalised linear model may result in overdispersion or underdispersion when does not display the right amount of variability for its mean value.

A Breakdown of the Gaussian Distribution

The exponential family form is

$$f(\pi; y) = \exp\left\{-\frac{(y^2 - 2y\mu + \mu^2)}{2\sigma^2} - \frac{1}{2}\log(2\pi\sigma^2)\right\}$$

so the various components are

$$\begin{aligned} &\psi = \sigma^2, && \theta = \mu, \qquad b(\theta) = \frac{1}{2}\theta^2, \\ &c(y;\psi) = -\frac{1}{2\psi}y^2 - \frac{1}{2}\log(2\pi\sigma^2). \end{aligned}$$

Then the canonical link function and the variance are

$$\begin{aligned} \mathbb{E}(Y) = \mu = \theta \qquad &\Rightarrow \qquad \text{the identity link, and} \\ \operatorname{Var}(Y) = \sigma^2 \qquad &\Rightarrow \qquad V(\mu) = 1. \end{aligned}$$

A Breakdown of the Binomial Distribution

The exponential family form is

$$f(\pi; y, n) = \exp\left\{ y \log \frac{\pi}{1-\pi} + n \log(1-\pi) + \log \binom{n}{y} \right\}$$

so the various components are

$$\psi = 1, \quad \theta = \log \frac{\pi}{1-\pi}, \quad b(\theta) = n \log(1 + e^{\theta}), \quad c(y; \psi) = \log \binom{n}{y}.$$

Then the canonical link function and the variance are

$$\begin{aligned} \mathbb{E}(Y) = \mu = n\pi = n\frac{e^{\theta}}{1+e^{\theta}} \qquad &\Rightarrow \qquad \theta = \log \frac{\pi}{1-\pi} \qquad \text{and} \\ \operatorname{Var}(Y) = \frac{e^{\theta}}{(1+e^{\theta})^2} \qquad &\Rightarrow \qquad V(\pi) = n\pi(1-\pi) \\ &\qquad\qquad\quad = n^{-1}\mu(n-\mu). \end{aligned}$$

A Breakdown of the Poisson Distribution

The exponential family form is

$$f(\lambda; y) = \exp\{y \log \lambda - \lambda - \log y!\}$$

so the various components are

$$\psi = 1, \qquad \theta = \log \lambda, \qquad b(\theta) = e^{\theta}, \qquad c(y; \psi) = -\log y!$$

Then the canonical link function and the variance are

$$\mathbb{E}(Y) = \lambda = e^{\theta} \qquad \Longrightarrow \qquad \theta = \log \lambda$$

and

$$\operatorname{Var}(Y) = e^{\theta} = \lambda \qquad \Longrightarrow \qquad V(\mu) = \mu.$$

Exponential Family and Maximum Likelihood Estimation

Maximum likelihood estimates for $\hat{\boldsymbol{\beta}}$ can be derived through iteratively (re-)weighted least squares (IWLS). Say $\mathbb{E}(Y_i) = \mu_i$. Then we use the chain rule a few times and string the resulting derivatives:

$$\begin{aligned}
\frac{\partial l}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial l_i}{\partial \beta_j} &= \sum_{i=1}^{n} \frac{\partial l(\theta_i)}{\partial \theta} \frac{\partial \theta(\mu_i)}{\partial \mu} \frac{\partial \mu(\eta_i)}{\partial \eta} \frac{\partial \eta(\beta_j)}{\partial \beta_j} \\
&= \sum_{i=1}^{n} \psi_i^{-1} \left(Y_i - b(\theta_i)\right) \left(\frac{\partial \mu(\theta_i)}{\partial \theta} \right)^{-1} \frac{1}{g'(\eta_i)} x_{ij} \\
&= \sum_{i=1}^{n} \frac{1}{w_i \psi} \frac{(Y_i - \mu_i)}{g'(\eta_i) b''(\theta_i)} x_{ij} \\
&= \sum_{i=1}^{n} \frac{W_i}{\psi} (Y_i - \mu_i) g'(\eta_i) x_{ij}. \qquad (10)
\end{aligned}$$

where $W_i^{-1} = w_i g'(\eta_i)^2 b''(\theta_i)$.

Second Derivative

The Fisher information is the matrix with entries

$$\mathbb{E}\left(-\frac{\partial^2 l}{\partial\beta_r\partial\beta_s}\right) = \sum_{i=1}^{n} W_i x_{ir} x_{is}, \tag{11}$$

i.e. $I(\boldsymbol{\beta}) = \boldsymbol{X}^T\boldsymbol{W}\boldsymbol{X}$ where $\boldsymbol{W}$ is a diagonal matrix with entries W_i.

We can use these derivatives to iteratively update the estimates of β_j with Newton-Raphson or Fisher scoring:

$$\begin{aligned} \boldsymbol{\beta}^{(t+1)} &= \boldsymbol{\beta}^{(t)} - I(\boldsymbol{\beta}^{(t)})^{-1}\,\nabla l(\boldsymbol{\beta}^{(t)}) \\ &\approx (\boldsymbol{X}^T\boldsymbol{W}^{(t)}\boldsymbol{X})^{-1}(\boldsymbol{X}^T\boldsymbol{W}^{(t)}\boldsymbol{Y}). \end{aligned} \tag{12}$$

The weights $W_i^{(t)}$ are re-evaluated after each iteration (hence IWLS).

Note that if g is the canonical link $(b')^{-1}$, then $b'' = (g')^{-1}$, so $W_i^{-1} = w_i g'(\eta_i)$.

Goodness of Fit: The Deviance

The main measure of goodness of fit is the deviance, which is twice the difference between the log-likelihoods of two nested models:

$$D = 2\left(l(\hat{\boldsymbol{\beta}}) - l(\tilde{\boldsymbol{\beta}})\right) = 2\psi^{-1}\sum_{i=1}^{n} w_i\left\{Y_i(\hat{\theta} - \tilde{\theta}) - b(\hat{\theta}) + b(\tilde{\theta})\right\}. \quad (13)$$

This is a log-likelihood ratio test statistic and so is asymptotically distributed as a χ^2-distribution whose degrees of freedom are given by the difference in number of the free parameters.
If ψ is unknown and has to be estimated, the result above still holds as long as we use a consistent estimate $\hat{\psi}$.
We can define the unscaled deviance

$$D^* = \psi D = \sum_{i=1}^{n} 2w_i\left\{Y_i(\hat{\theta} - \tilde{\theta}) - b(\hat{\theta}) + b(\tilde{\theta})\right\}. \quad (14)$$

for distributions with a meaningful dispersion parameter.

The Null and Residual Deviance

The two most basic forms of deviance used in model selection are:

- The null deviance

  $$D_N = 2\left[l(\mathcal{M}_S) - l(\mathcal{M}_0)\right] \sim \chi^2_{n-1},$$

  comparing the saturated model $\mathcal{M}_S$ and the model with just an intercept $\mathcal{M}_0$. It is useful for comparing fit quality, but don't focus on it too much.

- The residual deviance

  $$D_R = 2\left[l(\mathcal{M}_S) - l(\mathcal{M}_L)\right] \sim \chi^2_{n-p-1},$$

  comparing the the saturated model $\mathcal{M}_S$ and the model $\mathcal{M}_L$ estimated from the data. For Gaussian GLMs, the residual deviance is (surprise!) the scaled residual sum of squares

  $$D_R = \sum_{i=1}^{n} \frac{(Y_i - \hat{\mu}_i)^2}{\sigma^2} \sim \chi^2_{n-p-1}.$$

Analysis of Deviance

Like the analysis of variance, analysis of deviance can be used to decompose the deviance of a full fitted model into independent components associated with the explanatory variables. Starting from the null deviance we can split out the residual deviance

$$D_N = 2\left[l(\mathcal{M}_S) - l(\mathcal{M}_0)\right] = 2\left[\underbrace{l(\mathcal{M}_S) - l(\mathcal{M}_L)}_{\text{residual deviance } D_R} + \underbrace{l(\mathcal{M}_L) - l(\mathcal{M}_0)}_{\text{model deviance}}\right],$$

and then we can split the component related to each explanatory variable:

$$D_N = D_R + 2\left[\underbrace{l(\mathcal{M}_L) - l(\mathcal{M}_{\beta_{p-1}})}_{\text{component for } \beta_p} + \underbrace{l(\mathcal{M}_{\beta_{p-1}}) - l(\mathcal{M}_{\beta_{p-2}})}_{\text{component for } \beta_{p-1}} + \right.$$

$$\left. + \ldots + \underbrace{l(\mathcal{M}_{\beta_1}) - l(\mathcal{M}_0)}_{\text{component for } \beta_1}\right].$$

Other Model Selection Criteria

Aside from using deviance instead of residual variance, model selection is mostly the same as for ordinary linear models.

- We can build tables like ANOVA tables with deviance contributions and χ^2 tests.
- We can use AIC and BIC to compare models that are not necessarily nested.
- For predictive models, we can use cross-validation to compute predictive correlations (for continuous responses), true positives & negatives (for binary responses) or classification errors (for categorical responses).

Model assumptions and goodness-of-fit should be checked in the process of selecting a model, to make sure the selected model makes sense.

Residuals

The definition of the residuals is more ambiguous than in the case of ordinary linear models because of the link function the different possible scales for prediction. Two common takes are:

- Pearson's residuals,

  $$\rho_i = \frac{Y_i - \hat{\mu}_i}{\sqrt{V(\hat{\mu}_i)}} \qquad \text{where } V(\mu_i) = \psi_i^{-1}\,\mathrm{Var}(Y_i) = b''(\theta_i), \tag{15}$$

  which are ordinary residuals standardised by the variance adjusted for the dispersion parameter. Note that if Y_i is Poisson then

  $$\sum_{i=1}^{n} \rho_i^2 = \text{Pearson's } X^2.$$

- the deviance residuals d_i, defined so that

  $$D = \sum_i d_i(Y_i, \hat{\mu}_i)^2 \qquad \mathrm{sign}(d_i) = \mathrm{sign}(Y_i - \hat{\mu}_i). \tag{16}$$

  and D is the deviance of the model.

Properties of the Residuals

- The residuals of a generalised linear model are not normally distributed. They should be used to look for violations of the mean and variance models.
- If the response is discrete, residuals usually appear in stripe patterns, with one stripe for each level of the response.
- For both definitions, the sum of the squared residuals is approximately distributed as a χ^2_{n-p-1}.
- Pearson's residuals have approximately zero mean and constant variance ψ but they can be quite skewed.
- Deviance residuals are more likely to look like they are normally distributed. Plus, they can be interpreted as the contribution of the ith observation to D.
- Model selection often tries to minimise deviance residuals, (almost) never Pearson's residuals.

Residuals vs Fitted Values: an Example

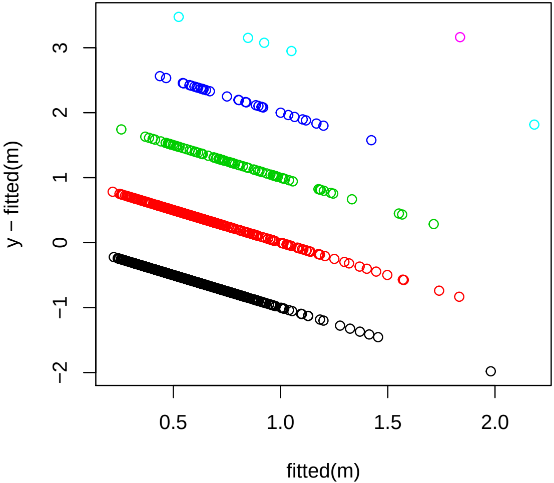
The stripe pattern in the residuals appears even if the model is perfectly specified, as the Poisson GLM below.

```
> set.seed(123)
> n = 10^3
> k = 5
> beta = rnorm(k, sd = 0.2)
> x = matrix(rnorm(n * k), ncol = k)
> y = rpois(n, lambda = exp(-0.5 + x %*% beta))
> m = glm(y ~ x, family = poisson)
```

To highlight which residual corresponds to which value of y, we can produce a custom plot with one colour for each observed value of the response.

```
> plot(fitted(m), y - fitted(m), col=y+1)
```

Residuals vs Fitted Values: an Example



Estimating the Dispersion Parameter

For the binomial and Poisson GLMs, $\psi = 1$ and therefore there is (in theory) nothing to estimate. For other distributions, the dispersion parameter is a nuisance parameter that we can estimate as:

$$\hat{\psi} = \frac{1}{n-p-1} \sum_{i=1}^{n} \frac{(Y_i - \hat{\mu}_i)^2}{V(\hat{\mu}_i)/w_i}, \tag{17}$$

which is the sample variance of the Pearson residuals.

For Gaussian GLMs, we obtain the unbiased estimate of the residual's variance:

$$\hat{\psi} = \sum_{i=1}^{n} \frac{(Y_i - \hat{\mu}_i)^2}{n-p-1} = \sum_{i=1}^{n} \frac{\hat{\varepsilon}_i^2}{n-p-1} = \hat{\sigma}_\varepsilon^2.$$

Note that using $n-p-1$ rather than n is to reduce finite sample bias, but the statistic is only unbiased in the ordinary linear model case.

Overdispersion

The downside of not having a dispersion to estimate is that the variance $\mathrm{Var}(Y)$ is purely a function of $\mathbb{E}(Y)$, and model estimation only cares about the latter. Therefore, the data may actually be more variability (overdispersion) or less variable (underdispersion).

The MLE does not depend upon ψ at all, but if it is misspecified then

- standard errors for $\boldsymbol{\beta}$ will be wrong;
- all the goodness-of-fit statistics are biased.

There are several model that extend GLMs to handle such data sets under more relaxed assumptions: the beta-binomial model, the gamma-Poisson model, quasi-likelihood models, random-effects models, double exponential families, etc.

How Do We Discover Overdispersion?

The common way of assessing overdispersion is to compare the residual deviance against its degrees of freedom, because the two quantities should be similar

Recall $D_R \simeq \chi^2_{n-p-1}$, so we should have $\mathbb{E}D_R \approx n-p-1$ with standard deviation around $\sqrt{2(n-p-1)}$. If the observed D_R is more than a couple of standard deviations away from the mean, then this is worthy of further investigation.

In practice, using the sum of the squared Pearson's residuals to check whether

$$\hat{\psi} = \frac{1}{n-p-1} \sum_{i=1}^{n} \rho_i^2 \simeq 1 \tag{18}$$

as opposed to using the deviance residuals d_i^2 has much less bias.

Again, under the model the quantity $(n-p-1)\hat{\psi} \simeq D_R$, so they both have a χ^2_{n-p-1} distribution.

The Anolis Lizards Data Set

This small data set is from Fienberg's (1980) book on categorical data analysis.

```
> lizards = read.table("lizards.txt", header = TRUE)
> head(lizards, 3)

  Species Diameter Height
1  Sagrei   narrow    low
2  Sagrei   narrow    low
3  Sagrei   narrow    low
```

For a sample of 409 lizards, the following variables were recorded:

- the species, which can be either `Sagrei` or `Distichus`;
- the diameter of the branch they were perched on, discretised in two categories `narrow` ($\leqslant 4$in) and `wide` (> 4in);
- the height of that same branch, discretised in two categories `high` (> 4.75ft) and `low` ($\leqslant 4.75$ft).

Fitting GLMs: the `glm()` Function

The `glm()` function is the analogue of `lm()` for GLMs, and has a similar syntax. The main difference lies in the family argument, which specifies the distribution we are assuming for the response and (optionally) the link function. The default is the canonical link.

```
> m = glm(Species ~ Diameter + Height, data = lizards,
+          family = binomial(link = logit))
> summary(m)
```

Let Y_{ijk} denote the species of the kth lizard from a branch with diameter $i \in \{1, 2\}$ (narrow, wide) and height $j \in \{1, 2\}$ (high, low).
$Y_{ijk} = 0$ denotes Distichus and $Y_{ijk} = 1$ Sagrei. The ordering of these definitions depends upon the ordering of the factor labels in R.

Here we're fitting the model $Y_{ijk} \sim \text{Binom}(1, \pi_{ijk})$ independently with

$$\text{logit}\,\pi_i = \mu + \alpha_i + \gamma_j.$$

R uses the corner point constraint $\alpha_1 = \gamma_1 = 0$ for identifiability.

Models From glm() and summary()

Here is the (edited) output from summary(m):

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8048  -1.1170   0.6609   0.9326   1.2390

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.1437     0.1503  -0.956 0.338972
Diameterwide   0.8029     0.2198   3.652 0.000260 ***
Heightlow      0.7511     0.2242   3.350 0.000807 ***

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 550.85  on 408  degrees of freedom
Residual deviance: 526.57  on 406  degrees of freedom
AIC: 532.57
```

summary(m): Regression Coefficients

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -0.1437     0.1503  -0.956 0.338972
Diameterwide  0.8029     0.2198   3.652 0.000260 ***
Heightlow     0.7511     0.2242   3.350 0.000807 ***
```

The p-values for the Wald tests are computed using z-scores (as opposed to the t-scores used for `lm()` models), which are defined as

$$z_{\beta_i} = \frac{\hat{\beta}_i}{se(\hat{\beta}_i)} \longrightarrow N(0,1) \qquad \text{under } H_0 : \beta_i = 0. \tag{19}$$

In a Gaussian linear model under the null the z-value is exactly t_{n-p-1}, but not in a GLM, so we fall back on the CLT. For example:

```
> 2*(1 - pnorm(0.8029/0.2198))

[1] 0.0002593293
```

The standard errors $se(\hat{\beta}_i)$ come from the IWLS.
Don't get hung up on hypothesis tests: effect sizes and confidence intervals are more important.

summary(m): Goodness of Fit

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 550.85  on 408  degrees of freedom
Residual deviance: 526.57  on 406  degrees of freedom
AIC: 532.57
```

The null and residual deviance are reported with the respective degrees of freedom. If $l(\mathcal{M}_S) = 0$, as in this case, we have

$$D = -2l(\mathcal{M}_L) \qquad \text{which means} \qquad \mathrm{AIC} = D + 2(p+1). \tag{20}$$

R^2 is not reported, because even though a few pseudo-R^2 coefficients have been defined they are difficult to interpret. Focussing on absolute (as opposed to relative) goodness-of-fit is generally misguided.

Key Quantities from glm()

- The fitted values $\hat{\mu}_i$, obtained by transforming the linear predictors by the inverse of the link function, i.e. $g^{-1}(\hat{\eta}_i)$.

```
> fitted(m)[1:5]  # can also use m$fitted.values

        1         2         3         4         5
0.3526623 0.3526623 0.3526623 0.3526623 0.3526623
```

- The residuals.

```
> resid(m, type="pearson")[1:5]  # default: deviance resids

       1        2        3        4        5
1.354833 1.354833 1.354833 1.354833 1.354833
```

- The intercept and regression coefficients $\hat{\boldsymbol{\beta}}$.

```
> coef(m) # or m$coefficients

 (Intercept) Diameterwide    Heightlow
   0.1437035   -0.8028584   -0.7510606
```

The predict() Function

predict() produces predicted or fitted values (if no newdata is passed to the function) on two scales:

- on the scale of the linear predictors, i.e. the $\hat{\eta}_i$;

```
> predict(m, type = "link")[1:5]

         1          2          3          4          5 
-0.6073571 -0.6073571 -0.6073571 -0.6073571 -0.6073571 
```

- on the scale of the response, i.e. $\hat{\mu}_i$;

```
> predict(m, type = "response")[1:5]

        1         2         3         4         5 
0.3526623 0.3526623 0.3526623 0.3526623 0.3526623 
```

For example, for a logistic regression the former returns $\hat{\eta}_i = \operatorname{logit} \hat{\pi}_i$ while the latter returns $\hat{\pi}_i$.

The argument se.fit=TRUE gives standard errors (using Fisher info).

Deviance Tables from anova()

anova() returns a table with the decomposition of the deviance, starting from the empty model $\mathcal{M}_0$ and adding each explanatory in turn in the order in which they were specified in the call to glm().

```
> anova(m)
Analysis of Deviance Table

Model: binomial, link: logit
Response: Species

Terms added sequentially (first to last)

         Df Deviance Resid. Df Resid. Dev
NULL                       408     550.85
Diameter  1   12.606       407     538.24
Height    1   11.674       406     526.57
```

The first entry is the null deviance (i.e. the residual deviance of $\mathcal{M}_0$) and the last is the residual deviance (i.e. of $\mathcal{M}_L$) from summary().

Leverage and Cook's Distance

The residual variance of a linear model is replaced by the dispersion ψ and the hat matrix becomes

$$\boldsymbol{H} = \boldsymbol{W}^{\frac{1}{2}} \boldsymbol{X} (\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{W}^{\frac{1}{2}},$$

with diagonal elements h_i a measure of influence (the leverage) of the ith observation. A rule of thumb is to worry if $h_i \gg p/n$ (e.g. $> 3p/n$). By arguments analogous to those used in ordinary linear models, the standardised, studentised residuals

$$\rho_i^* = \frac{(Y_i - \hat{\mu}_i)}{\sqrt{\hat{\psi} V(\hat{\mu}_i)(1 - h_{ii})}}, \qquad d_i^* = \frac{d_i}{\sqrt{\hat{\psi}(1 - h_{ii})}}.$$

should have roughly constant variance and may be used to identify outliers.
An alternative is the Cook's distances; calculation for GLMs is computationally challenging and they are usually approximated.

Model Checking

Checking the model is much more difficult for a GLM than for an ordinary linear model. Diagnostic plots depend on the nature of the response and of the explanatory variables, particularly the residuals' quantile-quantile plot.

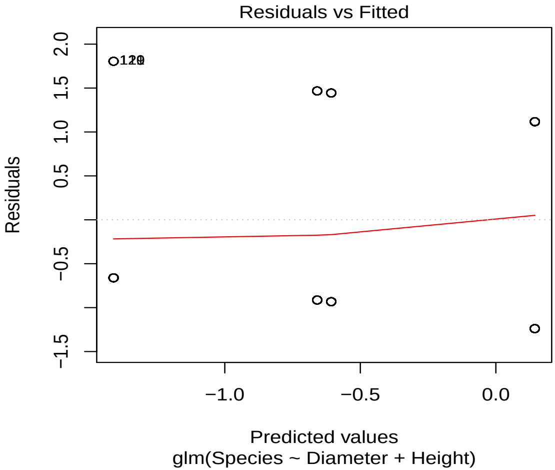
The main assumptions to check are that the mean and variance models are correctly specified.

To check the mean model, one can plot residuals against fitted values (R uses the deviance residuals and $\hat{\eta}_i$). No trend should be observed if the model is correct.

For the variance one can plot a standardised residual against a fitted value (R uses $\sqrt{|d_i^*|}$ against $\hat{\eta}_i$). Again we shouldn't see any trend. We can equally replace $\hat{\eta}_i$ with $\hat{\mu}_i$ as preferred, and look at Pearson residuals instead of deviance residuals; however Pearson residuals are generally more skewed, which may make inspection more difficult.

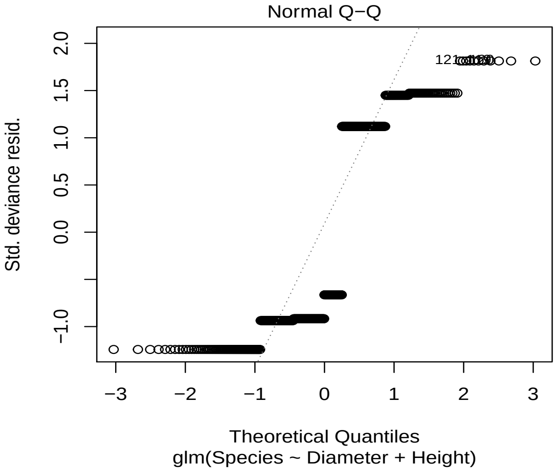
Graphical Diagnostics: `plot(m, which=1)`

The first plot is of d_i against $\hat{\eta}_i$ for checking the mean model. No discernible trend.



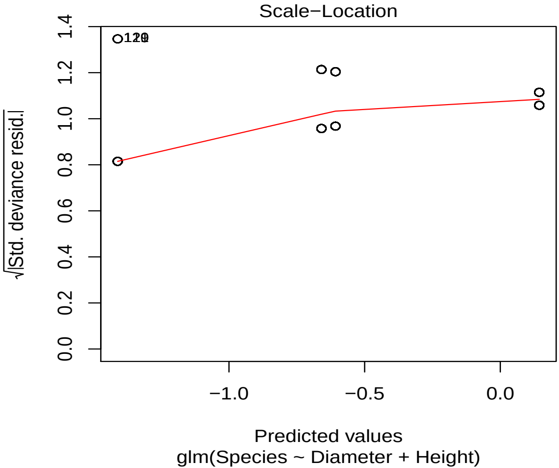
Graphical Diagnostics: `plot(m, which=2)`

The next is a normal Q-Q plot of the deviance residuals. There is no reason to expect these to be normal, but it may help to identify outliers.



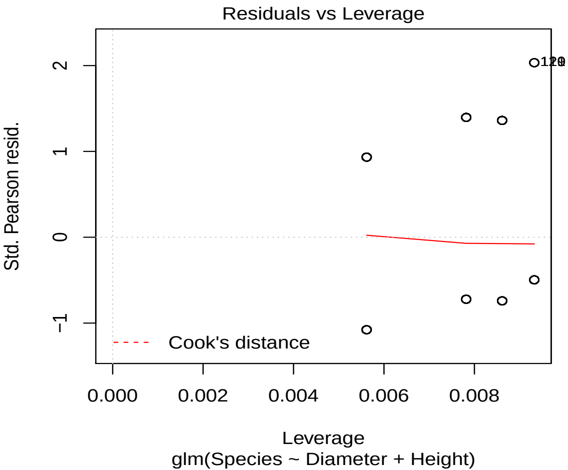
Graphical Diagnostics: `plot(m, which=3)`

The next is of $\sqrt{|d_i^*|}$ against $\hat{\eta}_i$, which is a check of the variance model. No discernible trend.



Graphical Diagnostics: `plot(m, which=5)`

The last plot is ρ_i^* against the leverage, and might help identify regions of strong influence in the covariate space.



Logistic Regression

Model Formulation

Logistic regression is a binomial GLM with the canonical logit link

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \eta_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p$$

which means that for each observation

$$\pi_i = \frac{\exp(\beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p)}{1 + \exp(\beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p)}.$$

The relationship is linear between the logarithm of the odds of success and the regressors. In other words, each regression coefficient represents the logarithm of the estimated change in the odds for a unit change of the corresponding explanatory variable.

Logistic regression is widely used in machine learning for classification because it scales well: see the Data Mining course.

Prostatic Cancer: an Epidemiological Study

This data set from Collett's "Binary Data Modelling" book describes an epidemiological study on the diagnosis of nodal involvement in prostatic cancer based on non-invasive methods. The study includes 53 patients and 5 explanatory variables:

- **Age:** the age of the patient, in years.
- **Acid:** the level of serum acid phosphate.
- **X-ray:** the result of x-ray examination, `positive` or `negative`.
- **Size:** tumour size, `small` or `large`.
- **Grade:** tumour grade, `less` or `more` serious.

```
> cancer = read.table("prostatic.cancer.txt", header = TRUE)
> head(cancer, 3)

  Age Acid     Xray  Size Grade Nodal
1  66 0.48 negative small  less    no
2  68 0.56 negative small  less    no
3  66 0.50 negative small  less    no
```

The Importance of Factor Coding

The coding of the factors involved in the logistic regression will affect the interpretion of the regression coefficients. In the case of the response variable, swapping cases and controls just changes the signs of all the regression coefficients because

$$-\log\left(\frac{\pi}{1-\pi}\right) = \log\left(\frac{1-\pi}{\pi}\right) = \log\left(\frac{\pi'}{1-\pi'}\right) \quad \text{with } \pi' = 1-\pi. \tag{21}$$

If relevant, the first level should correspond to controls and the second to cases.

As for the explanatory variables, contrasts are built using the first level as a reference so the regression coefficients may or may not be easily interpreted depending on which is chosen. We can take care of that with the `relevel()` function.

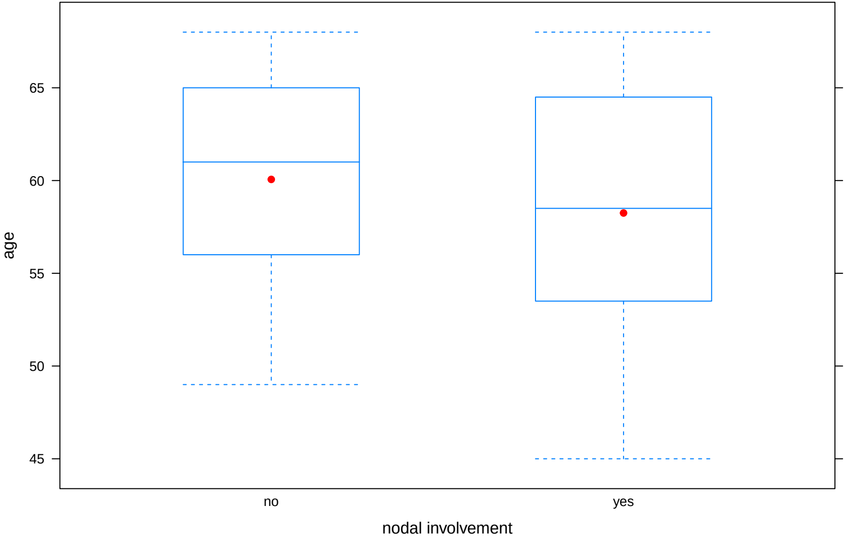
```
> cancer$Nodal = relevel(cancer$Nodal, ref = "no")
> cancer$Grade = relevel(cancer$Grade, ref = "less")
> cancer$Size = relevel(cancer$Size, ref = "small")
> cancer$Xray = relevel(cancer$Xray, ref = "negative")
```

Graphical Methods for Exploratory Analysis

Graphical exploratory analysis techniques developed for ordinary linear models are unsuited to logistic regression because they implicitly assume the response is continuous. Three alternatives are:

- Plotting each continuous explanatory variable against the response using boxplots.
- Plotting each categorical explanatory variable against the response using mosaic plots.
- Plotting the first principal component for the explanatory variables against the second in a principal components plot. Cases and controls are in different colours and should ideally cluster.

Boxplots for Continuous Explanatory Variables



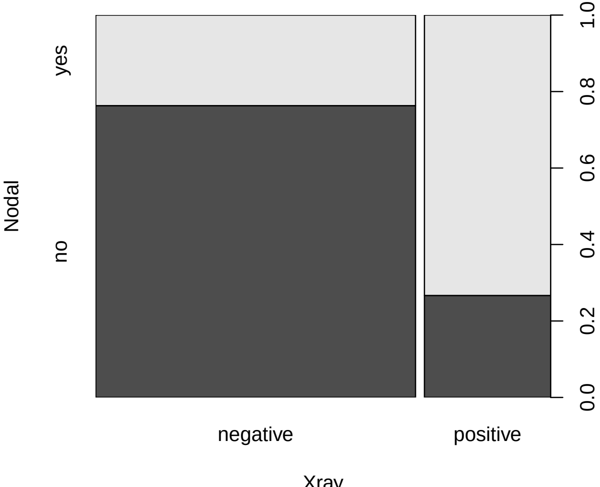
Boxplots for Continuous Explanatory Variables (R Code)

```
> library(lattice)
> bwplot(Age ~ Nodal, data = cancer,
+        xlab = "nodal involvement", ylab = "age",
+        panel = function(x, y, ...) {

+          panel.bwplot(x, y, ..., pch = "|")
+          panel.points(1, mean(y[x == "no"]), pch = 19, col = "red", ...)
+          panel.points(2, mean(y[x == "yes"]), pch = 19, col = "red", ...)
+        }
+ )
```

Stacked Barplots for Categorical Explanatory Variables

```
> plot(Nodal ~ Xray, data=cancer)
```



Fitting the Logistic Regression Model

Let's fit the model with everything.

```
> m  = glm(Nodal ~ I(Age/10) + Acid + Xray + Size + Grade,
+          data = cancer, family = binomial)
> summary(m)
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.6259     3.4598   0.470   0.6384
I(Age/10)     -0.6926     0.5788  -1.197   0.2314
Acid           2.4344     1.3158   1.850   0.0643 .
Xraypositive   2.0453     0.8072   2.534   0.0113 *
Sizesmall     -1.5641     0.7740  -2.021   0.0433 *
Grademore      0.7614     0.7708   0.988   0.3232
---

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 70.252  on 52  degrees of freedom
Residual deviance: 48.126  on 47  degrees of freedom
AIC: 60.126
```

Model

Being explicit, the model we're fitting on the previous slide is $Y_i \sim \mathrm{Binom}(1, \pi_i)$ independently where:

$$\mathrm{logit}(\pi_i) = \beta_0 + \beta_1 \frac{\mathsf{age}_i}{10} + \beta_2 \mathsf{acid}_i + \beta_3 \mathbb{1}_{\{\mathsf{Xray}_i = \mathsf{positive}\}} + \cdots$$
$$\cdots + \beta_4 \mathbb{1}_{\{\mathsf{Size}_i = \mathsf{large}\}} + \beta_5 \mathbb{1}_{\{\mathsf{Grade}_i = \mathsf{more}\}}$$

Note that in our model, β_1 is the change in log-odds of nodal cancer associated with a 10 year increase in age. This is often better than a 1 year increase, which is likely to have a very small effect (if any).

It's often useful to pick a baseline which corresponds to a realistic parameter value. In our model, β_0 is the log-odds of nodal cancer for a person aged 0, but the age range in the sample is 45–68.

Pick units that correspond to a meaningful and comprehensible scale.

Coefficients and Confidence Intervals

Consider the line

```
               Estimate Std. Error z value Pr(>|z|)
 I(Age/10)      -0.6926     0.5788  -1.197   0.2314
```

This means $\hat{\beta}_1 = -0.69$ with standard error 0.58. This suggests a 95% confidence interval of

$$\hat{\beta}_1 \pm z_{1-\alpha/2} se(\hat{\beta}_1) \qquad = (-1.8, 0.44).$$

Do not report spurious decimal places!

How should we interpret this? Well,

$$\log \frac{P(Y_i = 1)}{P(Y_i = 0)} = \eta_i = \beta_0 + \cdots + x_{ip}\beta_p,$$

so $\hat{\beta}_j$ is the (estimated) change in the log-odds of the cancer being nodal for a 1 unit increase in x_{ij}.

See AA Chapter 5 for more on parameter interpretation.

Interpreting Odds

e^{β_1} is the multiplicative change in the odds of the cancer being nodal. We have

$$e^{\hat{\beta}_1} = 0.50 \; (0.16, 1.6)$$

So each 10 years of age is associated with a halving of the odds of nodal cancer, but it could be as much as a 60% increase or an 84% decrease!

This is easist to think about if $P(Y = 1)$ is small, in which case

$$\text{odds}(Y = 1) \approx P(Y = 1).$$

Reporting actual probabilities is often good for interpretation.

Odds Ratios

If the jth covariate is binary, then the coefficient β_j corresponds to the log odds ratio between Y_i and x_{ij} (conditional on the other covariates being fixed).

One way to estimate the (observed) effect of one variable (X) on another (Y) is to look at how the odds of Y occuring change when we condition on different levels of X. One way of measuring this is the ratio of those odds:

$$\begin{aligned} OR(X, Y) &= \frac{P(Y = 1 \mid X = 1)}{P(Y = 0 \mid X = 1)} \Big/ \frac{P(Y = 1 \mid X = 0)}{P(Y = 0 \mid X = 0)} \\ &= \frac{P(Y = 1 \mid X = 1) \cdot P(Y = 0 \mid X = 0)}{P(Y = 0 \mid X = 1) \cdot P(Y = 1 \mid X = 0)}. \end{aligned}$$

It's not hard to see that this is symmetric in X and Y, so in fact $OR(X, Y) = OR(Y, X)$, and it can be derived from the distribution of $P(X = x \,|\, Y = y)$ in the same way.

Odds Ratios

Odds ratios can be hard to interpret; these all correspond to an odds ratio of about 8:

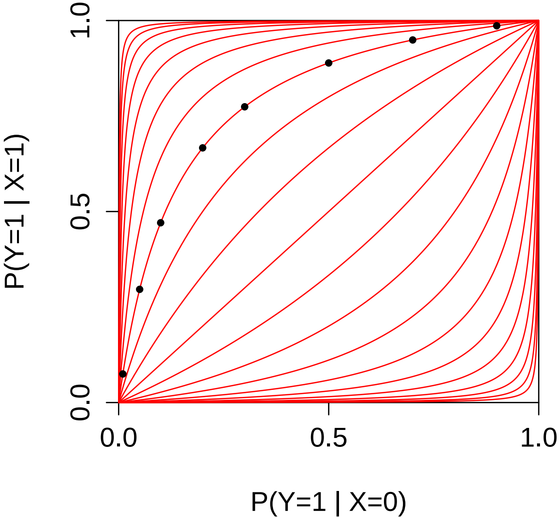
$P(Y = 1 \mid X = 0)$	0.01	0.05	0.1	0.2	0.3	0.5	0.7	0.9
$P(Y = 1 \mid X = 1)$	0.07	0.30	0.47	0.67	0.77	0.89	0.95	0.99

If $P(Y = 1 \mid X = x)$ small then it approximates the risk ratio:

$$OR(X, Y) \approx RR(X, Y) = \frac{P(Y = 1 \mid X = 1)}{P(Y = 1 \mid X = 0)}.$$

The risk ratio is generally easier to explain.

L'Abbé Plot for Lines of Constant Ratio



Parameter Interpretation: Collapsibility

Consider two logistic regression models with linear components:

$$\eta_i = \beta_0 + \beta_1 x_i + \beta_2 z_i \qquad\qquad \eta_i^* = \beta_0^* + \beta_1^* x_i.$$

- β_1^* is change in log-odds with 1 unit change in x_i (i.e. a marginal measure of association);
- β_1 is change in log-odds with 1 unit change in x_i and z_i held constant (i.e. a conditonal measure of association);
- $\beta_1 \neq \beta_1^*$.

So far so obvious, but unlike ordinary linear regression (and also log-linear regression), this is true even if there is no relationship between x_i and z_i. This is because (for continuous variables) if $\operatorname{logit} \pi_i = \eta_i$ is correct then $\operatorname{logit} \pi_i = \eta_i^*$ is generally incorrect. Logistic regression is non-collapsible. This means two things:

- the distinction between marginal and conditional parameters is particularly important;
- don't believe your logistic regression model! (all models are wrong, etc...)

Parameter Interpretation: Interactions

As with a linear model we can add in interaction terms to improve the fit:

$$\operatorname{logit} P(Y_i = 1) = \beta_0 + \beta_1 x_i + \beta_2 z_i + \beta_3 x_i z_i.$$

Interpretation of interactions is even more difficult than first order parameters, and some caution is needed.

The presence of an interaction is scale dependent: change your link function and it may appear or disappear. It usually has no intrinsic meaning in terms of the original data.

A Note: The Hauck-Donner Phenomenon

Testing regression coefficients with deviance tests is more reliable than using the corresponding Wald tests due to a paradox called the Hauck - Donner phenomenon. What happens is that as the distance between the $\hat{\beta}_j$ and the null value increases, the test statistic decreases to 0.

This means, counter-intuitively, that we might fail to reject the null hypothesis because the effect of an explanatory variable is "too significant"!

This can happen when there is perfect or near-perfect separation of successes and failures in terms of an explanatory variable. Then the $\sqrt{\mathrm{Var}(\hat{\beta}_j)} \to \infty$ faster than $\hat{\beta}_j \to \infty$, so the z-score tends to zero.

Case-Control Studies

A case-control study is a kind of epidemiological study in which subjects are sampled based on the value of the binary response variable (Y) of interest.

So, to study lung cancer we might recruit 500 patients with lung cancer (the cases), and 500 patients without lung cancer (the controls). We can then compare the differences between the covariates ($\boldsymbol{X}$) of the two groups (e.g. to find risk factors for lung cancer).
This is especially useful for rare diseases, when prospective sampling will yield few (or no) cases.

Rather than sampling from the population distribution, say $P(\boldsymbol{X}, Y)$, we sample from a distribution

$$P^*(\boldsymbol{X}, Y) \equiv P(\boldsymbol{X} \mid Y) \cdot P^*(Y),$$

where the marginal $P^*(Y)$ is chosen by study design; the conditional distribution of the covariates given the response is unchanged.

Invariance of Logistic Regression

Remarkably, if $P(Y \mid \boldsymbol{X})$ follows the logistic regression model

$$\operatorname{logit} P(Y = 1 \mid \boldsymbol{x}) = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p,$$

then so does $P^*(Y \mid \boldsymbol{X})$, with only the intercept changed:

$$\operatorname{logit} P^*(Y = 1 \mid \boldsymbol{x}) = \beta_0^* + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p.$$

This is because odds ratios are invariant to marginal distributions.

As a consequence logistic regression is very commonly used in epidemiology, because it's often much more convenient to sample cases and controls than to sample prospectively.

Note that the 'baseline' prevalence e^{β_0} can't be estimated because of the sampling, so we can only recover the odds ratios (not odds or probabilities).

Note that for rare diseases the odds ratios are approximately the same as risk ratios, so we can interpret the coefficients as multiplicative effects (as in the log-linear models we will see later).

predict(m): Classify Cases and Controls

Logistic regression can be used to classify individuals as cases or controls. This can be done in R with `predict()`, either from the data used to fit the model `m` or from new data.

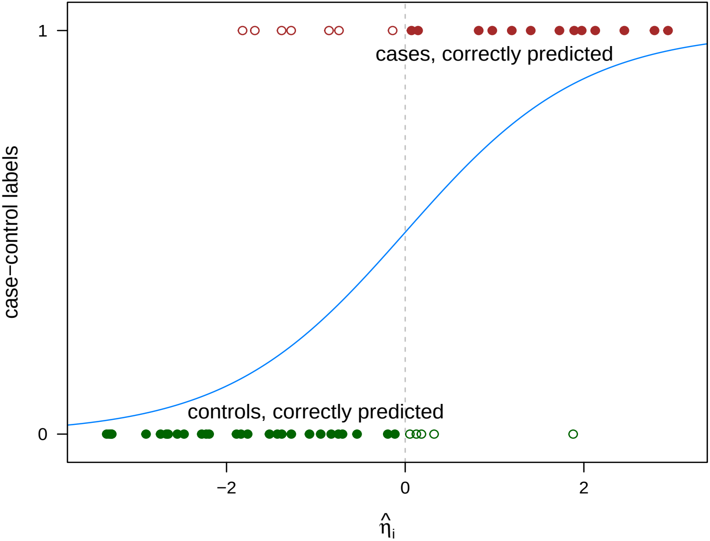
```
> PRED = ifelse(predict(m, type = "response") >= 0.5, "yes", "no")
> table(OBS = cancer$Nodal, PRED)

     PRED
OBS   no yes
  no  28   5
  yes  7  13
```

The four cells in the table above, which is called a confusion matrix, indicate how many observations are correctly identified as cases or controls by the model:

- Cases with a predicted $\hat{\pi}_i \geqslant 0.5$ are true positives (TP).
- Cases with $\hat{\pi}_i < 0.5$ are false negatives (FN).
- Controls with $\hat{\pi}_i \geqslant 0.5$ are false positives (FP).
- Controls with $\hat{\pi}_i < 0.5$ are true negatives (TN).

Plotting Fitted and Observed Responses



Plotting Fitted and Observed Responses (R Code)

```
> logit <- function(x) log(x/(1-x))       # logit function
> expit <- function(x) exp(x)/(1+exp(x))  # expit function
> library(lattice)
> col = trellis.par.get()$superpose.symbol$col[c(3, 7)]
>
> xyplot(as.numeric(cancer$Nodal) - 1 ~ logit(fitted(m)),
+   xlab = expression(hat(eta)[i]), ylab = "case-control labels",
+   scales = list(y = list(at = c(0, 1)), tck = c(1, 0)),
+   panel = function(x, y, ...) {

+     panel.xyplot(x, y, col = col[y + 1],
+         pch = c(19, 1)[((y == 0) & (x > 0)) |
+                        ((y == 1) & (x < 0))) + 1])
+     panel.abline(v = 0, col = "grey", lty = 2)
+     panel.text(x = -1, y = 0.1, pos = 1, "controls, correctly predicted"
+     panel.text(x = 1, y = 0.9, pos = 3, "cases, correctly predicted")
+     sq <- seq(min(x)-.5, max(x)+.5, length.out=1000)
+     panel.xyplot(sq, expit(sq), type = "l")
+   })
```

Accuracy, Sensitivity and Specificity

The goodness of fit and predictive ability of logistic regression (as well as other binary classification models) are measured using various functions of TP, TN, FP and FN. Say the number of cases is $\mathrm{P} = \mathrm{TP} + \mathrm{FN}$ and the number of controls is $\mathrm{N} = \mathrm{TN} + \mathrm{FP}$.

The first of these measures is the accuracy, the proportions of observations that are correctly classified:

$$\mathrm{ACCURACY} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}} = \frac{\text{observations that are correctly classified}}{\text{sample size}}.$$

Then there are sensitivity,

$$\mathrm{SENSITIVITY} = \frac{\mathrm{TP}}{\mathrm{P}} = \frac{\text{observations correctly classified as cases}}{\text{number of cases}};$$

and specificity

$$\mathrm{SPECIFICITY} = \frac{\mathrm{TN}}{\mathrm{N}} = \frac{\text{observations correctly classified as controls}}{\text{number of controls}}.$$

Precision and Recall

Another set of measures are precision

$$\text{PRECISION} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{\text{observations correctly classified as cases}}{\text{observations classified as cases}}$$

and recall, which is another name for sensitivity.

To add to the confusion, sensitivity is also called the true positive rate (TPR) and specificity is also called the true negative rate (TNR). This naming convention is the same as in multiple testing adjustment, where we try to control the false positive rate (FPR) through the false discovery rate (FDR):

$$\text{FPR} = \frac{\text{FP}}{\text{N}} \qquad \text{and} \qquad \text{FDR} = \frac{\text{FP}}{\text{TP} + \text{FP}} = 1 - \text{PRECISION}.$$

Computing Them From the Confusion Matrix

So, in the confusion matrix we have

```
> tab = table(OBS = cancer$Nodal, PRED)
> TN = tab[OBS = "no", PRED = "no"]
> FN = tab[OBS = "yes", PRED = "no"]
> FP = tab[OBS = "no", PRED = "yes"]
> TP = tab[OBS = "yes", PRED = "yes"]
```

and then we can compute

```
> (accuracy = (TP + TN) / nrow(cancer))

[1] 0.7735849

> (sensitivity = TP / (TP + FN))

[1] 0.65

> (specificity = TN / (TN + FP))

[1] 0.8484848
```

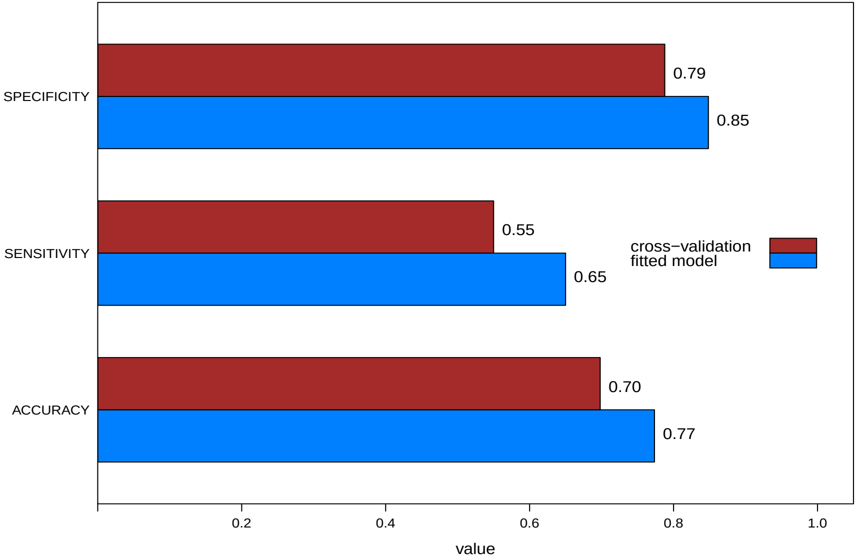
All these measures are defined in $[0, 1]$, and high values are assigned to good models which fit or predict the data well.

Computing Them Using Cross-Validation

```
> # shuffle the data to get unbiased splits.
> kcv = split(sample(nrow(cancer)), seq_len(10))
>
> predicted = lapply(kcv, function(test) {
+   dtraining = cancer[-test, ]   # training data
+   dtest = cancer[test, ]        # rest is test
+   model = glm(Nodal ~ Age + Acid + Xray + Size + Grade, data = dtraining,
+               family = binomial(link = "logit"))  # fit to training
+   # predict the data in the test data.
+   PRED = ifelse(predict(model, newdata = dtest, type = "response") >= 0.5,
+                  "yes", "no")
+   # return the observed-predicted pairs.
+   return(data.frame(OBS = dtest$Nodal, PRED = PRED))
+ })
>
> # collate all the predictions from the different folds.
> predicted = do.call("rbind", predicted)
> table(predicted)

     PRED
OBS   no yes
  no  26   7
  yes  8  12
```

Comparing Goodness of Fit and Predictive Power



Comparing Goodness of Fit and Predictive Power (R Code)

```
> d = data.frame(
+   MEASURE = rep(c("ACCURACY", "SENSITIVITY", "SPECIFICITY"), 2),
+   MODEL = c(rep("FITTED", 3), rep("XVAL", 3)),
+   VALUE = c(0.7735, 0.65,0.8484,0.6981,0.55,0.7878))
>
> library(lattice)
> col = trellis.par.get()$superpose.symbol$col[c(1, 7)]
>
> barchart(MEASURE ~ VALUE, group = MODEL, data = d,
+     xlim = c(0, 1.05), xlab = "value", scales = list(tck = c(1, 0)),
+     auto.key = list(corner = c(0.95, 0.5), points = FALSE, rectangles = TRUE,
+                     text = c("fitted model", "cross-validation"),
+                     reverse.rows = TRUE),
+     par.settings = simpleTheme(col = col),
+     panel = function(x, y, groups, ...) {

+       panel.barchart(x, y, groups = groups, ...)
+       panel.text(x = x,
+                  y = as.numeric(y) + ((as.numeric(groups) - 1.5) * 2) * 0.15,
+                  labels = sprintf("%.2f", x), pos = 4)

+     })
```

Predictive Power and The Bias-Variance Trade-Off

Logistic regression is susceptible to overfitting, just as are ordinary linear models. Symptoms are markedly reduced values for sensitivity, specificity and accuracy in cross-validation compared to the model fitted on the whole data.

Some caution is needed in reasoning on these quantities. For example, note that when the sample is very unbalanced (i.e. very few cases compared to controls):

- specificity is inflated, because there are so many N that all models will have a high TN and thus $\mathrm{TN}/\mathrm{N} \to 1$;
- accuracy is similarly inflated, because TN dominates TP so $\mathrm{TN} + \mathrm{TP} \simeq \mathrm{TN}$ for all models;
- sensitivity loses discriminatory power, because TP/P is defined in increments of $1/\mathrm{P}$.

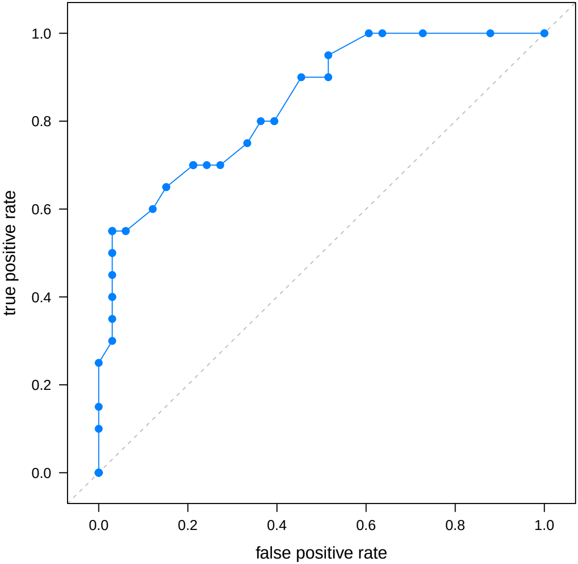
ROC Curves

A graphical, synthetic summary of such classification goodness-of-fit measure is the receiver operating characteristic (ROC) curve. Model performance is represented as a curve of sensitivity (the true positive rate) against $1 - \text{SPECIFICITY}$ (the false positive rate). The curve is bounded in $[0, 1] \times [0, 1]$, the ROC space:

- A perfect classification model would be in $(0, 1)$ because it has sensitivity 1 (no false negatives) and specificity 1 (no false positives).
- A model that is equivalent to a random guess would be on the diagonal, since then Ps and Ns are equally likely to be classified as P.
- Models above the diagonal are good classifiers, models below are worse than random.

The curve is produced by varying the discrimination threshold that determines whether an observation is classified as a case or not; for logistic regression the default is $\hat{\pi}_i \geqslant 0.5$. This can be done either in the context of model fitting or in cross-validation.

The ROC Curve for the Logistic Regression Model



Building a ROC Curve (R Code)

Building a ROC curve on the whole data entails fitting the model once, building the confusion matrix and then varying the value of $\hat{\pi}_i$.

```
> m  = glm(Nodal ~ Age + Acid + Xray + Size + Grade, data = cancer,
+          family = binomial(link = "logit"))
> roc = data.frame(x = numeric(41), y = numeric(41))
> # 40 thresholds in 2.5% increments.
> thr = seq(from = 0, to = 1, by = 0.025)
> for (i in seq_along(thr)) {
+   PRED = ifelse(predict(m, type = "response") >= thr[i], "yes", "no")
+   PRED = factor(PRED, levels = c("no", "yes"))
+   tab = table(OBS = cancer$Nodal, PRED, useNA = "always")
+   # compute false positive rate.
+   roc[i, "x"] =  tab[OBS = "no", PRED = "yes"] / sum(tab[OBS = "no", ])
+   # compute true positive rate.
+   roc[i, "y"] =  tab[OBS = "yes", PRED = "yes"] / sum(tab[OBS = "yes", ])
+ }#FOR
```

Doing the same from cross-validated predictions works in the same way; unless multiple runs of cross-validation can be used to produce averaged coordinates for each `thr` for improved stability.

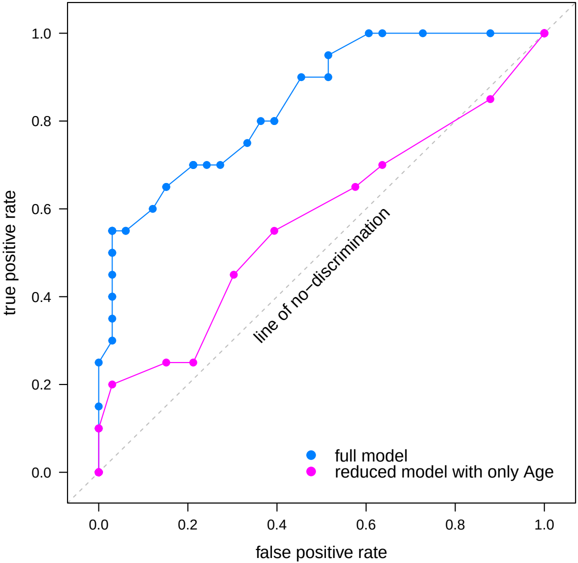
Building and Interpreting a ROC Curve

- It is tricky to guess how many values of the threshold are needed to obtain a smooth-ish curve, because neither axis is a direct function of the threshold. This is important if the model takes time to fit and/or cross-validation is run many times.

- Models can be compared but it is unlikely any of them will strictly dominate the others over the whole ROC space. The closer a ROC curve is to the left and upper bounds of the ROC space, the better classifier is the corresponding model.

- All curves start at $(0, 0)$ and end up at $(1, 1)$, and any reasonable model for binary responses should produce curves that are strictly above the diagonal.

Comparing ROC Curves



A Summary Statistic for ROC Curves

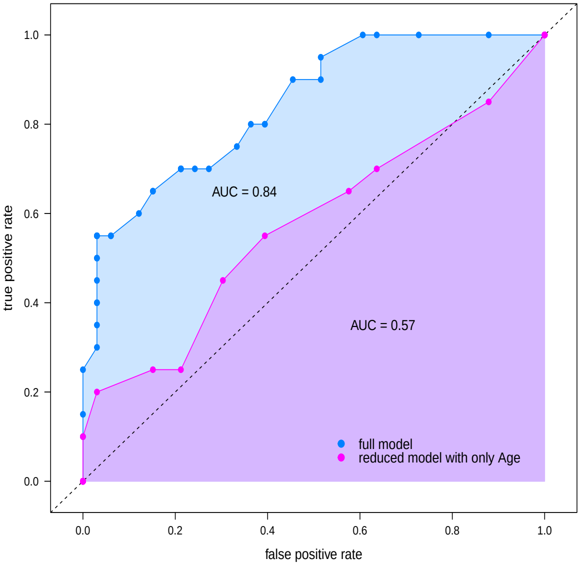
Clearly comparing models through their ROC curves is a principled approach, but it does not scale well to large number of models and it is ambiguous when the curves overlap and cross each other.

A popular summary statistic for a ROC curve is the area under the curve (AUC). If the curve is above the diagonal it ranges from 0.5 (e.g. the model does not perform any better than picking at random) and 1 (e.g. perfect classifier). An informal evaluation scale is:

from	**to**	**interpretation**
0	0.60	Bad
0.61	0.70	Acceptable
0.71	0.80	Good
0.81	1	Excellent

and 0.75 is a rough threshold for classification accuracy on cross-validated predictions.

Comparing AUC Values



Comparing AUC Values (R Code)

```
> m2  = glm(Nodal ~ Age, data = cancer, family = binomial(link = "logit"))
> roc = data.frame(x = numeric(82), y = numeric(82),
+          model = c(rep("M1", 41), rep("M2", 41)))
> ## ...
> ## compute ROC values as above, and then...
```

```
> xyplot(y ~ x, groups = model, data = roc, type = "b",
+    scales = list(tck = c(1, 0)),
+    xlab = "false positive rate", ylab = "true positive rate", pch = 19,
+    key = list(points = list(pch = 19, col = col), corner = c(0.85, 0.10),
+            text = list(c("full model", "reduced model with only Age"))),
+    panel = function(x, y, groups, ...) {

+       panel.polygon(x = c(1, x[groups == "M1"]), y = c(0, y[groups == "M1"]),
+         col = col[1], border = col[1], alpha = 0.2)
+      panel.polygon(x = c(1, x[groups == "M2"]), y = c(0, y[groups == "M2"]),
+        col = col[2], border = col[2], alpha = 0.2)
+      panel.abline(0, 1, lty = 2, col = "black")
+      panel.xyplot(x, y, groups, ...)
+      panel.text(x = 0.65, y = 0.35, "AUC = 0.57")
+      panel.text(x = 0.35, y = 0.65, "AUC = 0.84")

+  })
```

Ranking Models by AUC Values

The AUC for a ROC curve can be easily approximated using the trapezoid method, which is a one-liner from the `roc` data frame:

```
> r1 = roc[roc$model == "M1", ]
> r2 = roc[roc$model == "M2", ]
> sum(abs(r1$x[2:41] - r1$x[1:40])*(r1$y[2:41] + r1$y[1:40])/2)

[1] 0.8424242

> sum(abs(r2$x[2:41] - r2$x[1:40])*(r2$y[2:41] + r2$y[1:40])/2)

[1] 0.5742424
```

We can then rank the models by AUC and use it as we would use AIC or BIC to select the best classifier. Unlike deviance tests, models need not to be nested. As usual, AUC has to be computed under cross-validation to select the best predictive model.

Note, however, that telling whether two AUC values are significantly different is an open problem without a widespread, accepted solution.

Log-Linear Regression

Model Formulation

Log-linear regression is a Poisson GLM with the canonical logarithmic link. That is, $Y_i \sim \mathrm{Pois}(\lambda_i)$ where

$$\log(\lambda_i) = \eta_i = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p \tag{22}$$

which means that for each observation

$$\lambda_i = \exp(\beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p)$$

The relationship is linear between the logarithm of the intensity (i.e. the expected count) and the regressors. In other words, each regression coefficient introduces a multiplicative contribution linked to changes in the corresponding explanatory variable: increasing x_{ij} by 1 whilst keeping other covariates fixed means

$$\lambda_i \mapsto \lambda_i e^{\beta_j}.$$

Species in the Galapagos: an Example from Ecology

This data set from Ramsey & Schafer's "Statistical Sleuth" book describes the number of native and non-native species in relation to:

- Island: name of the island.
- Total number of species and number of native species.
- Area (km^2) and Elevation (m).
- Distance from the nearest island (DistNear) and from Santa Cruz (DistSc).
- AreaNear: area of the nearest island.

```
> galapagos = read.table("galapagos.txt", header = TRUE)
> head(galapagos, 3)

     Island Total Native  Area Elev DistNear DistSc AreaNear
1    Baltra    58     23 25.09  332      0.6    0.6     1.84
2 Bartolome    31     21  1.24  109      0.6   26.3   572.33
3  Caldwell     3      3  0.21  114      2.8   58.7     0.78
```

Modelling the Total Number of Species

We can use step() to perform step-wise selection by AIC.

```
> m0 = glm(Total ~ 1, data = galapagos, family = poisson)
> m1 = step(m0, scope = ~ log(Area) + log(Elev)
+                         + log(DistNear) + log(AreaNear))
```

```
Start:  AIC=3673.56; Total ~ 1
[...]
Step:  AIC=552.09;   Total ~ log(Area) + log(AreaNear) + log(DistNear)

> summary(m1)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    3.378320   0.048865   69.14  < 2e-16 ***
log(Area)      0.366261   0.008227   44.52  < 2e-16 ***
log(AreaNear) -0.099160   0.006143  -16.14  < 2e-16 ***
log(DistNear) -0.059823   0.011707   -5.11 3.22e-07 ***
---
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 3510.73  on 29  degrees of freedom
Residual deviance:  383.26  on 26  degrees of freedom
```

Huge Residual Deviance?

Why is the residual deviance (383.26) so large compared to the residual degrees of freedom (26)? It is due to overdispersion:

```
> psi <- sum(residuals(m1, type = "pearson")^2)/m1$df.res
> psi

[1] 16.16604
```

Remember that $\psi = 1$, supposedly.

If this happens then we have various options, but do not ignore overdispersion!

- Try adding more covariates to get a better fit: don't do this mindlessly, however.
- Use a quasi-Poisson model in order to get reasonable standard errors (essentially just scale them by $\sqrt{\hat{\psi}}$). See next section.
- Do something else entirely.

A Link Between Logistic and Log-Linear Models

Exercise. Let $N = Y_1 + Y_2$. Then one can show

$$\left.\begin{array}{l} Y_1 \sim \mathrm{Pois}(\lambda_1) \\ Y_2 \sim \mathrm{Pois}(\lambda_2) \end{array}\right\} \text{ independently } \iff \left\{\begin{array}{l} N \sim \mathrm{Pois}(\lambda_1 + \lambda_2) \\ Y_1 \mid N \sim \mathrm{Binom}\left(N, \frac{\lambda_1}{\lambda_1+\lambda_2}\right) \end{array}\right.$$

So if we have, for example, $Y_{ij} \sim \mathrm{Pois}(\lambda_{ij})$ with means $\log \lambda_{ij} = \mu + \alpha_i + \beta_j + \gamma_{ij}$ then

$$Y_{2j} \mid Y_{+j} \sim \mathrm{Binom}(Y_{+j}, \pi_j)$$

where $\operatorname{logit} \pi_j = \alpha_2 - \alpha_1 + \gamma_{2j} - \gamma_{1j}$.

Using the corner point $\alpha_1 = \gamma_{1j} = 0$, the parameters α_2 and γ_{22} are shared between the two models.

The parameters μ and β_j control the total number of trials, which is assumed fixed under the binomial model.
See the contingency tables lectures for more on this.

A Link Between Logistic and Log-Linear Models

Using the previous slide, when all the explanatory variables are categorical, the response can be summarised as a count for each of their configurations and then modelled as a Poisson random variable.
The plyr library gives methods for transforming data in this way.

```
> library(plyr)
> liz2 <- ddply(lizards, .variables = 1:3, nrow)
> liz2

    Species Diameter Height V1
1 Distichus   narrow   high 73
2 Distichus   narrow    low 61
3 Distichus     wide   high 70
4 Distichus     wide    low 41
5    Sagrei   narrow   high 86
6    Sagrei   narrow    low 32
7    Sagrei     wide   high 35
8    Sagrei     wide    low 11
```

A Link Between Logistic and Log-Linear Models

```
> m_logit <- glm(Species ~ Diameter, data=lizards, family=binomial
> m_pois <- glm(V1 ~ Species*Diameter, data=liz2, family=poisson)
```

Looking at the respective summaries:

```
> summary(m_logit)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.1272     0.1262  -1.007 0.313826
Diameterwide  -0.7537     0.2161  -3.488 0.000486 ***

> summary(m_pois)

Coefficients:
                          Estimate Std. Error z value Pr(>|z|)
(Intercept)                4.20469    0.08639  48.673  < 2e-16 ***
SpeciesSagrei             -0.12716    0.12624  -1.007 0.313824
Diameterwide              -0.18831    0.12834  -1.467 0.142309
SpeciesSagrei:Diameterwide -0.75373    0.21607  -3.488 0.000486 ***
```

Rates and Offsets

Poisson distributions are usually characterised as modelling the "number of events occurring in a fixed interval of time and/or space."

What if the data are not equally balanced? For example, suppose we count the number of road accidents Y_i in different cities i over different periods of time t_i? We would expect that $\mathbb{E}Y_i \propto t_i$ holding other variables constant.

In the Poisson case, we can write

$$\log \mathbb{E}Y_i = \log t_i + \log \mu_i$$

where μ_i is a measure of intensity per unit time.

We can proceed to model $\log \mu_i = \beta_0 + \cdots + x_{ip}\beta_p$ as before (though note it may change the parameters' interpretation).

$$\log(\lambda_i) - \log(n_i) = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p \tag{23}$$

to have an offset that expresses the exposure e.g. the length of the time interval in which events were counted or the number of subjects in the study. It must be a known constant.

The Poisson GLM then models an set of observation-specific intensities,

$$\log\left(\frac{\lambda_i}{n_i}\right) = \beta_0^* + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p. \tag{24}$$

Offset Example: Hodgkin's Lymphoma Data

The data set `hodgkins` contains the number of deaths by age and sex from Hodgkin's Lymphoma recorded in California in 1989.

```
> hodg <- read.table("hodgkins.txt", header=TRUE)
> head(hodg, 3)

    Age Sex Deaths     Pop
1 3034   M     50 1299868
2 3539   M     49 1240595
3 4044   M     38 1045453
```

There are both more deaths and more individuals at younger age groups, so an analysis which ignores population doesn't make sense.

Let's model the intensity per million of population against sex.

$Y_i \sim \text{Pois}(\lambda_i)$ with

$$\log\left(\frac{\lambda_i}{n_i/10^6}\right) = \beta_0 + \beta_1 \mathbb{I}_{\{\mathsf{Sex}_i = \mathsf{M}\}}.$$

Offset Example: Hodgkin's Lymphoma Data

Note that looking at rates per million of population makes numbers easier to think about.

```
> mod1 <- glm(Deaths ~ Sex + offset(I(log(Pop/1e6))),
+                               family=poisson, data=hodg)
> summary(mod1)
```

```
            Estimate Std. Error z value Pr(>|z|)    
(Intercept)  3.08780    0.07392   41.77  < 2e-16 ***
SexM         0.54377    0.09473    5.74 9.45e-09 ***

Residual deviance: 29.667  on 22  degrees of freedom
```

So the rate of lymphoma per million women is $e^{3.09} = 22$ with 95% CI $(19, 25)$, increasing to $e^{3.09+0.54} = 38$ for men (need to use Fisher Info to get the SE on this).
The model fit isn't great, but can be improved by adding a slope for age.

Offset Example: Hodgkin's Lymphoma Data

```
> age <- as.integer(hodg$Age)
> mod2 <- glm(Deaths ~ age + Sex + offset(I(log(Pop/1e6))),
+                   family=poisson, data=hodg)
> summary(mod2)
```

```
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 3.048085   0.106126  28.721  < 2e-16 ***
age         0.007975   0.015161   0.526    0.599
SexM        0.547999   0.095079   5.764 8.23e-09 ***

Residual deviance: 29.392  on 21  degrees of freedom
```

Rate of disease incidence appears not to be significantly related to age. However, adding a quadratic term suggests a significant U-shaped relationship, with the old and young most at risk.
(Naïve interpretation of risk in old age is likely to be erroneous for these data because of competing risks.)

Advanced Models

Quasi-Likelihood Models

From (10) we have that the score equations for the β_j can be written as

$$\frac{\partial l}{\partial \beta_j} = \sum_{i=1}^{n} \frac{(Y_i - \mu_i)}{\psi V(\mu_i)} \frac{\partial \mu(\eta_i)}{\partial \eta} x_{ij} = 0 \qquad \text{for all } j = 1, \ldots, p \tag{25}$$

to emphasise that the GLM depends on the distribution assumed for the response only through the mean and variance models.

So, we can do without any distributional assumption and switch to a model in which we only assume a form for $\mu_i(\boldsymbol{\beta})$ and

$$\mathrm{Var}(Y_i) = \psi V(\mu_i), \tag{26}$$

and solving (25), regardless of whether this is the derivative of a log-likelihood or not.

This is called a quasi-likelihood model.

Quasi-Likelihood Models

Quasi-Likelihood models are more flexible, but at the same time revert back to the usual GLM if the form of the mean and variance coincide with a parametric model.
In the Poisson GLM the variance is determined by $V(\mu_i) = \mu_i$ but in a quasi-likelihood model we take the more flexible

$$V(\mu_i) = \psi\mu_i$$

to model overdispersion (typically, it is assumed that $\psi \geqslant 1$). Similarly, in a binomial quasi-likelihood model we can assume

$$V(\pi_i) = \psi n_i \pi_i(1-\pi_i).$$

In both cases ψ drops out the likelihood equations, much like the residual variance does in Gaussian regression models, so parameter estimates are identical to that of Poisson and binomial GLMs. However standard errors are proportional to $\sqrt{\hat{\psi}}$.

A Quasi-Poisson Model for Hodgkin's

The syntax of `glm()` has `family=quasipoisson`; regression coefficient estimates are unchanged, but standard errors are rescaled by $\sqrt{\hat{\psi}}$.
The AIC is `NA` because the likelihood is undefined. The deviance shown is from the Poisson model, and is not scaled by $\hat{\psi}$ in the output; rescaling almost inevitably makes it close to the residual degrees of freedom.

```
> qm2 <- glm(Deaths ~ age + Sex + offset(I(Pop/1e6)),
+                                family=quasipoisson, data=hodg)
> summary(qm2)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.22757    0.11463  19.433 6.66e-15 ***
age         -0.05004    0.01621  -3.087 0.005587 **
SexM         0.47882    0.10785   4.440 0.000227 ***
---
(Dispersion parameter for quasipoisson family taken to be 1.29524)

    Null deviance: 67.777  on 23  degrees of freedom
Residual deviance: 27.728  on 21  degrees of freedom
AIC: NA
```

The Inevitability of Overdispersion

Let $Y_i \sim \text{Pois}(\lambda_i)$ where $\log \lambda_i = X_i + Z_i$, and X_i, Z_is are i.i.d. from some distribution.
Then $\mathbb{E}[Y_i \mid X_i, Z_i] = \mathrm{Var}[Y_i \mid X_i, Z_i] = \lambda_i$.

However, suppose we only measure X_i, but not Z_i. Then

$$\begin{aligned} \mathbb{E}[Y_i \mid X_i] &= \mathbb{E}[\mathbb{E}[Y_i \mid X_i, Z_i] \mid X_i] = e^{X_i}\mathbb{E}e^{Z_i} \\ \text{and} \quad \mathrm{Var}[Y_i \mid X_i] &= \mathrm{Var}\left(\mathbb{E}[Y_i \mid X_i, Z_i] | X_i\right) + \mathbb{E}\left[\mathrm{Var}(Y_i \mid X_i, Z_i) | X_i\right] \\ &= e^{2X_i}\,\mathrm{Var}\, e^{Z_i} + e^{X_i}\mathbb{E}e^{Z_i} \\ &= e^{X_i}\mathbb{E}e^{Z_i}\left(1 + e^{X_i}\frac{\mathrm{Var}\, e^{Z_i}}{\mathbb{E}e^{Z_i}}\right) \geq \mathbb{E}[Y_i \mid X_i], \end{aligned}$$

with equality if and only if the Z_i are constant.

Hence the data will be overdispersed if we fail to measure any explanatory variable. For this reason it's generally a good idea to use quasi-Poisson standard errors if any overdispersion is observed.

A Quasi-Binomial Model for Prostatic Cancer

It is actually possible to have a $\hat{\psi} < 1$ if the covariates are correlated but not measured (e.g. suppose you fix 50% men and 50% women in a trial), but it is more common that $\hat{\psi} \geqslant 1$. Here $\hat{\psi} = 1$ for practical purposes.

```
> summary(glm(Nodal ~ I(Age/10) + Acid + Xray + Size + Grade,
+                   data = cancer, family = quasibinomial))
```

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.6259     3.4521   0.471   0.6398
I(Age/10)     -0.6926     0.5775  -1.199   0.2364
Acid           2.4344     1.3129   1.854   0.0700 .
Xraypositive   2.0453     0.8054   2.540   0.0145 *
Sizesmall     -1.5641     0.7723  -2.025   0.0485 *
Grademore      0.7614     0.7691   0.990   0.3272

(Dispersion parameter for quasibinomial family taken to be 0.9955437)

    Null deviance: 70.252  on 52  degrees of freedom
Residual deviance: 48.126  on 47  degrees of freedom
AIC: NA
```

Zero-Inflated Poisson Model

Suppose you have a response variable representing counts, which would ideally be well fitted by a Poisson GLM. However, the data contains $Y_i = 0$ abnormally often due to the nature of the underlying phenomenon.

To account for this we can re-formulate the model as

$$Y_i \sim \begin{cases} 0 & \text{with probability } \pi \\ \text{Pois}(\lambda) & \text{with probability } 1 - \pi. \end{cases}$$

or equivalently

$$P(Y_i = 0) = \pi + (1 - \pi)e^{-\lambda}, \qquad P(Y_i = y_i) = (1 - \pi)\frac{\lambda^{y_i} e^{-\lambda}}{y_i!}.$$

This is known as zero-inflated Poisson (ZIP) model.

Mixture Model, or Bayesian Model?

From a frequentist point of view, this model is a mixture model combining a Poisson distributions and a Dirichlet mass at zero. Each observation is generated by one of these two distributions (the components of the mixture), but we do not know which. Therefore, model estimation is treated as a missing data problem in which there is an unobserved auxiliary dummy variable encoding which that missing information.

From a Bayesian point of view, we have a Poisson likelihood and we assign a prior distribution to λ with π as hyperparameter:

$$f(\lambda; \pi) = 0 \cdot \pi + \lambda \cdot (1 - \pi).$$

That is a spike-and-slab prior: it combines a diffuse probability distribution (the $\mathrm{Pois}(\lambda)$) with a point probability mass (spike at 0).

Estimates for λ and π

The methods of moments gives closed form estimates for both λ and π:

$$\hat{\lambda}_{\mathrm{MO}} = \frac{s^2 + \bar{Y}^2 - \bar{Y}}{\bar{Y}} \qquad \text{and} \qquad \hat{\pi}_{\mathrm{MO}} = \frac{s^2 - \bar{Y}}{s^2 + \bar{Y}^2 - \bar{Y}}$$

where s^2 is the sample variance.

The maximum likelihood solution is not in closed form, but λ can be estimated numerically by solving

$$\bar{Y}(1 - e^{-\lambda}) = \lambda \left(1 - \frac{n_0}{n}\right)$$

where n_0 is the number of observed zeros; and then that estimate can be plugged in

$$\hat{\pi}_{\mathrm{ML}} = 1 - \frac{\bar{Y}}{\hat{\lambda}_{\mathrm{ML}}}.$$

to estimate π.

Zero-Inflated Poisson Regression

In the context of regression, the ZIP model takes the form

$$\lambda_i = 0 \cdot \pi_i + \exp(x_{i1}\beta_1 + \ldots + x_{ip}\beta_p) \cdot (1 - \pi_i)$$

with

$$\begin{aligned} \log(\lambda_i) &= \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p \\ \log\left(\frac{\pi_i}{1-\pi_i}\right) &= \gamma_0 + z_{i1}\gamma_1 + \ldots + z_{iq}\gamma_q \end{aligned}$$

where the explanatory variables $x_{i1}, \ldots, x_{ip}$ for λ_i may or many not be the same as (or overlap with) the explanatory variables $z_{i1}, \ldots, z_{ip}$ for π_i. Estimation is performed as a missing data problem with the expectation-maximisation (EM) algorithm.

The `pcsi` library in R has code for this.

Another Model for Overdispersion: the Beta-Binomial

The beta-binomial model is the frequentist take on the classic Bayesian conjugate model

$$Y_i \mid \pi_i \sim \mathrm{Binom}(n_i, \pi_i) \qquad \text{with} \qquad \pi_i \sim \mathrm{Beta}(\alpha, \beta)$$

resulting in the compound model

$$P(Y_i = y \mid n_i, \alpha, \beta) = \binom{y}{n} \frac{\Gamma(y+\alpha)\Gamma(n-y+\beta)}{\Gamma(n+\alpha+\beta)} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$$

after integrating out π_i. The expected value and variance are

$$\begin{aligned} \mathbb{E}(Y_i) &= \frac{n\alpha}{\alpha+\beta} \equiv n\pi, \\ \mathrm{Var}(Y) &= \frac{n\alpha\beta(\alpha+\beta+n)}{(\alpha+\beta)^2(\alpha+\beta+1)} = n\pi(1-\pi)\frac{\alpha+\beta+n}{\alpha+\beta+1}. \end{aligned}$$

The Dispersion Parameter in the Beta-Binomial

The variance of the beta-binomial can be re-written as

$$\mathrm{Var}(Y) = n\pi(1-\pi)\frac{\alpha+\beta+n}{\alpha+\beta+1} = n\pi(1-\pi)[1+(n-1)\rho].$$

where ρ is effectively an over-dispersion parameter equal to

$$\rho = \frac{1}{\alpha+\beta+1}.$$

- If $\rho \to 0$ there is no overdispersion because $\mathrm{Var}(Y) = n\pi(1-\pi)$;
- if $\rho \to 1$ then $\mathrm{Var}(Y) \gg n\pi(1-\pi)$;
- the overdispersion is proportional to $n-1$, unlike in the quasi-Binomial model. In particular, it does not allow overdispersion for Bernoulli trials.

Note that it is not possible to model under-dispersion in this way.

Beta-Binomial Regression

It is possible to use a logistic link function to write beta-binomial regression model akin to binomial GLM,

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_0 + x_{i1}\beta_1 + \ldots + x_{ip}\beta_p \tag{27}$$

and estimate the α and β hyperparameters through π_i and the dispersion parameter ψ. However, even though the model can be fitted by maximising the likelihood numerically, it does not have all the favourable properties of GLMs because the beta-binomial distribution is not part of the exponential family.

For this reason a quasi-likelihood approach using the same variance function may be preferred.

See AA Chapter 14.

Negative Binomial Model

The beta-binomial model is a mixture model: specifically a Beta mix of Bernoulli distributions.
An analogous approach to dealing with overdispersion in the Poisson case is to use a gamma mixture of Poissons:

$$Y_i \sim \mathrm{Pois}(\lambda_i), \qquad \lambda_i \sim \Gamma(a, b)$$

This leads to a negative Binomial model with parameters k, p:

$$P(Y_i = y) = \binom{y + k - 1}{y} (1 - p)^k p^y, \qquad y = 0, 1, 2, \ldots,$$

where $p = (1 + b)^{-1}$ and $k = a$.
In this case

$$\mathbb{E} Y_i = \frac{a}{b}, \qquad \mathrm{Var}\, Y_i = \frac{a}{b}\left(1 + \frac{1}{b}\right),$$

so again only overdispersion is possible.

The Negative Binomial Distribution

We can rewrite the PMF for a negative binomial as

$$P(Y_i = y) = \frac{\Gamma(y+k)}{\Gamma(k)y!}(1-p)^k p^y, \qquad y = 0, 1, 2, \ldots,$$

extending the parameter space to any real $k > 0$.

If k is fixed, this is an exponential family with canonical parameter $\log(kp)$, so we can use it in a GLM.

We obtain the variance function

$$V(\mu) = \mu + \frac{1}{k}\mu^2 = \mathrm{Var}(Y).$$

The 'dispersion parameter' is 1, but really overdispersion is taken care of by estimating k.

Negative Binomial GLMs in R

The link functinos (and hence parameter interpretation) are generally the same as for the Poisson. A function to fit it is provided in the MASS package, with default log-link:

```
> library(MASS)
> summary(glm.nb(Total ~ log(Area) + log(DistNear) + log(AreaNear),
+                  data=galapagos))
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)    
(Intercept)   3.41362    0.17737  19.246   <2e-16 ***
log(Area)     0.35196    0.03733   9.427   <2e-16 ***
log(DistNear) -0.11381    0.07676  -1.483    0.138    
log(AreaNear) -0.04549    0.03843  -1.184    0.236    

(Dispersion parameter for Negative Binomial(2.4998) family taken to be 1)

Residual deviance:  33.071  on 26  degrees of freedom
```

The fits are similar to the quasi-Poisson case, though the scale-location looks slightly better for the (quasi-)Poisson.

Penalised Generalised Linear Models

Penalised regression in its standard form was introduced for linear models as penalised least squares optimisation, e.g.

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta})^T (\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_p \right\}, \qquad \lambda \geqslant 0;$$

$p = 1$ is the lasso, $p = 2$ is ridge regression. We can generalise this to a penalised maximum (log-)likelihood problem, which may also be applied to GLMs as e.g.

$$\underset{\boldsymbol{\beta}}{\operatorname{argmax}} \left\{ \sum_{i=1}^{n} l(\beta; Y_i) - \lambda \|\boldsymbol{\beta}\|_p \right\}, \qquad \lambda \geqslant 0$$

to produce the ridge and lasso equivalents for generalised linear models.

Note that if $p \geq 1$ this is a convex optimisation, and it scales reasonably well. See the book by Hastie, Tibshirani and Friedman (2009).

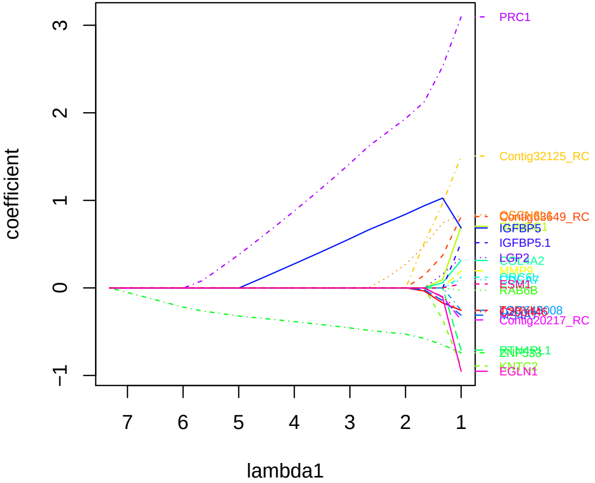
Penalised GLMs in R

The `penalized` package in R implements penalised GLMs.

```
> library(penalized)
> data(nki70)
> pen <- penalized(nki70[,2], unpenalized=nki70$Age,
+                  penalized=nki70[,8:77], model="logistic",
+                  lambda1=1.5, steps=20)
> plotpath(pen)
```

Naturally, some method is necessary for selecting a tuning parameter, and this can be done by cross validation using `cvl()`, for example.

Plot of the Coefficient Paths



Bayesian Approaches

Many of the hierarchical models used for overdispersion lend themselves very naturally to Bayesian approaches.

$$Y_i \mid \boldsymbol{\beta} \sim \mathsf{EDF}(\mu_i) \qquad g(\mu_i) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$$
$$\boldsymbol{\beta} \sim p_0(\boldsymbol{\beta})$$

In one sense there is little more to say: specify a prior for $\boldsymbol{\beta}$ and turn the crank. Common choices for priors are independent normals:

$$\beta_j \sim N(\mu_j, \sigma_j^2), \qquad j = 0, \ldots, p.$$

Pick the mean and variance to reflect plausible values for the coefficients. Note that this may be much easier if you rescale covariates appropriately: for example, make sure that intercept terms correspond to a plausible baseline covariate vector (e.g. not age 0 in the prostate cancer example).
See Gelman et al. *Bayesian Data Analysis*, Third Edition, 2013, for more on Bayesian approaches.

Hierarchical Models

Yet another way of modelling overdispersion is to assume that the parameter values are different for each individual and generated from some specific distribution.

For example:

$$Y_i \sim \mathrm{Pois}(\mu_i) \qquad\qquad \log \mu_i = \beta_0 + b_{0i} + (\beta_1 + b_{1i})x_i$$

where

$$\begin{pmatrix} b_{0i} \\ b_{1i} \end{pmatrix} \sim N\left(\mathbf{0}, \begin{pmatrix} \sigma_0^2 & \rho\sigma_0\sigma_1 \\ \rho\sigma_0\sigma_1 & \sigma_1^2 \end{pmatrix}\right).$$

This is form of mixed effects model, and can be fitted using either frequentist or Bayesian methods.

See AA Chapter 13 for details and more references.

Generalised Estimating Equations

Mixed effects models and hierarchical models allow for structured correlation between observations, but are likelihood based. Quasi-likelihood assumes independence, but doesn't require full likelihood specification.

Generalised Estimating Equations (GEE) methods combine some advantages of both, though at some loss of efficiency compared to likelihood-based methods. They solve an equation of the form

$$\sum_i \boldsymbol{D}_i^T \, \boldsymbol{V}_i(\boldsymbol{\alpha})^{-1} \left\{ \boldsymbol{Y}_i - \boldsymbol{\mu}_i(\boldsymbol{\beta}) \right\} = \boldsymbol{0}, \tag{28}$$

for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. Here $\boldsymbol{V}_i(\boldsymbol{\alpha})$ is a covariance matrix for $\boldsymbol{Y}_i$, and $\boldsymbol{D}_i = \nabla_{\boldsymbol{\beta}} \boldsymbol{\mu}_i$. The covariance structure can, for example, include exchangable or AR(1) observations.

This usually proceeds by alternating a fixed working covariance value $\hat{\boldsymbol{\alpha}}$, solving for $\hat{\boldsymbol{\beta}}$ and iterating.

See the R packages `geepack` and `lme4` and AA Chapter 12 for more details.

Fin