Generalized score tests

An idea that didn't work (although it worked in GLIM)

David Clayton and Johanna Howson

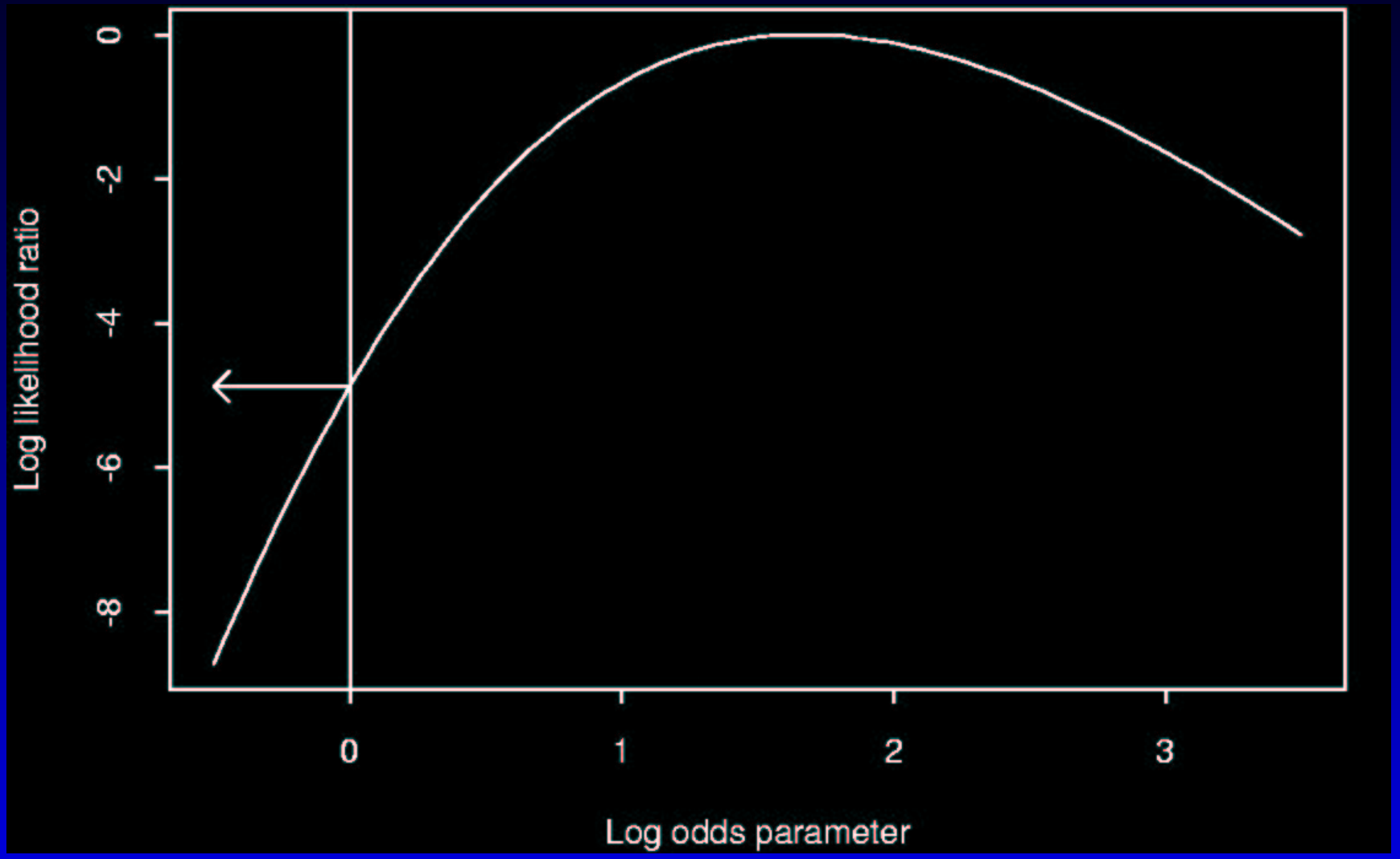
Diabetes and Inflammation Laboratory
Cambridge Institute for Medical Research,
Department of Medical Genetics, Cambridge

www-gene.cimr.cam.ac.uk/clayton

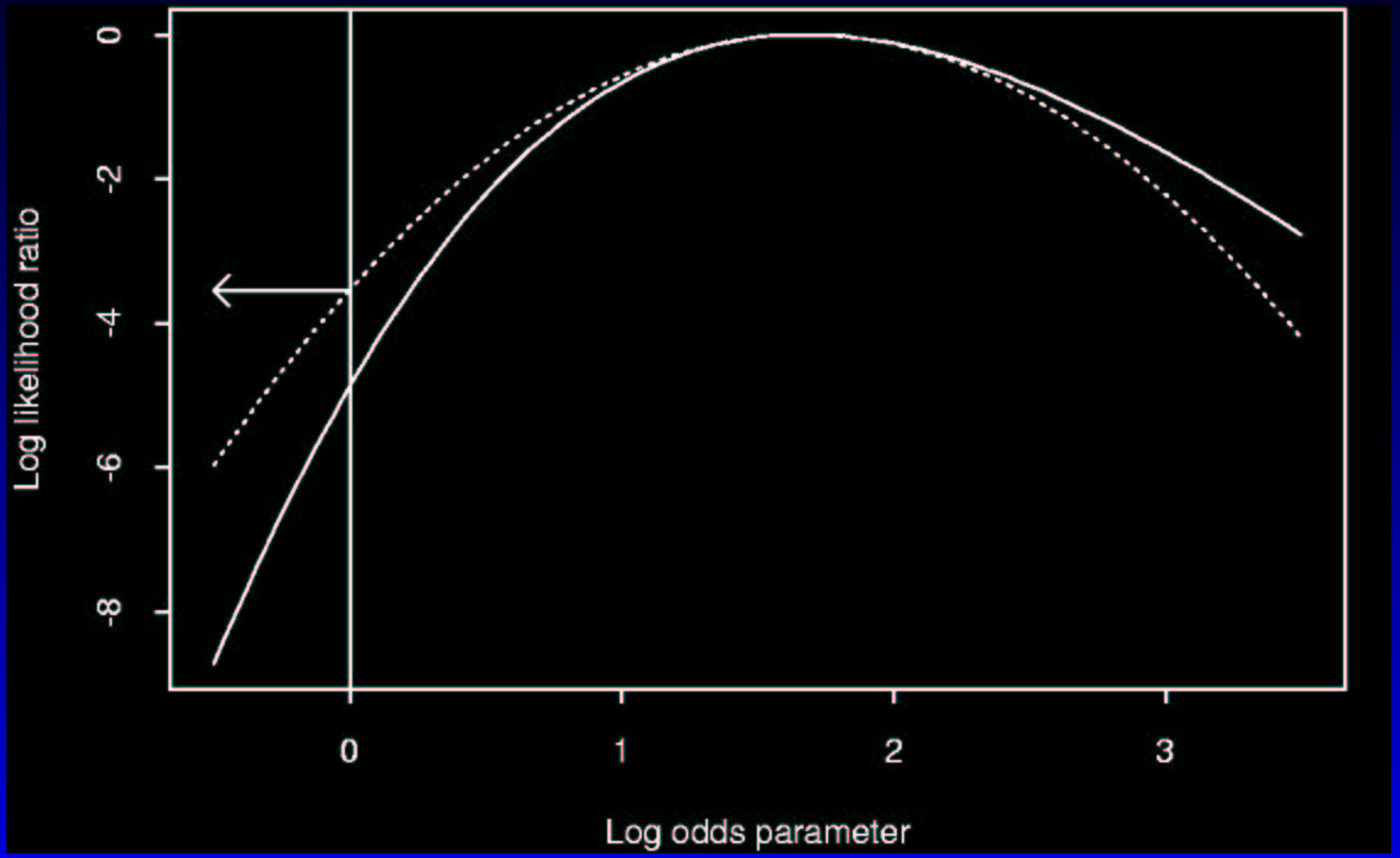
Tests based on likelihood ratios

- There are three general approaches to testing hypotheses concerning nested models:
  - Likelihood ratio test — compare log-likelihood at null model with its (maximized) value under the alternative
  - Wald test — compares difference between ML estimates of a group of parameters and their null value in relation to their variance (estimated from the inverse of the observed information)
  - Score test — tests the vector of gradients of the log-likelihood, evaluated at the null hypothesis
- The second two may be regarded as based on a quadratic approximation to the log–likelihood.

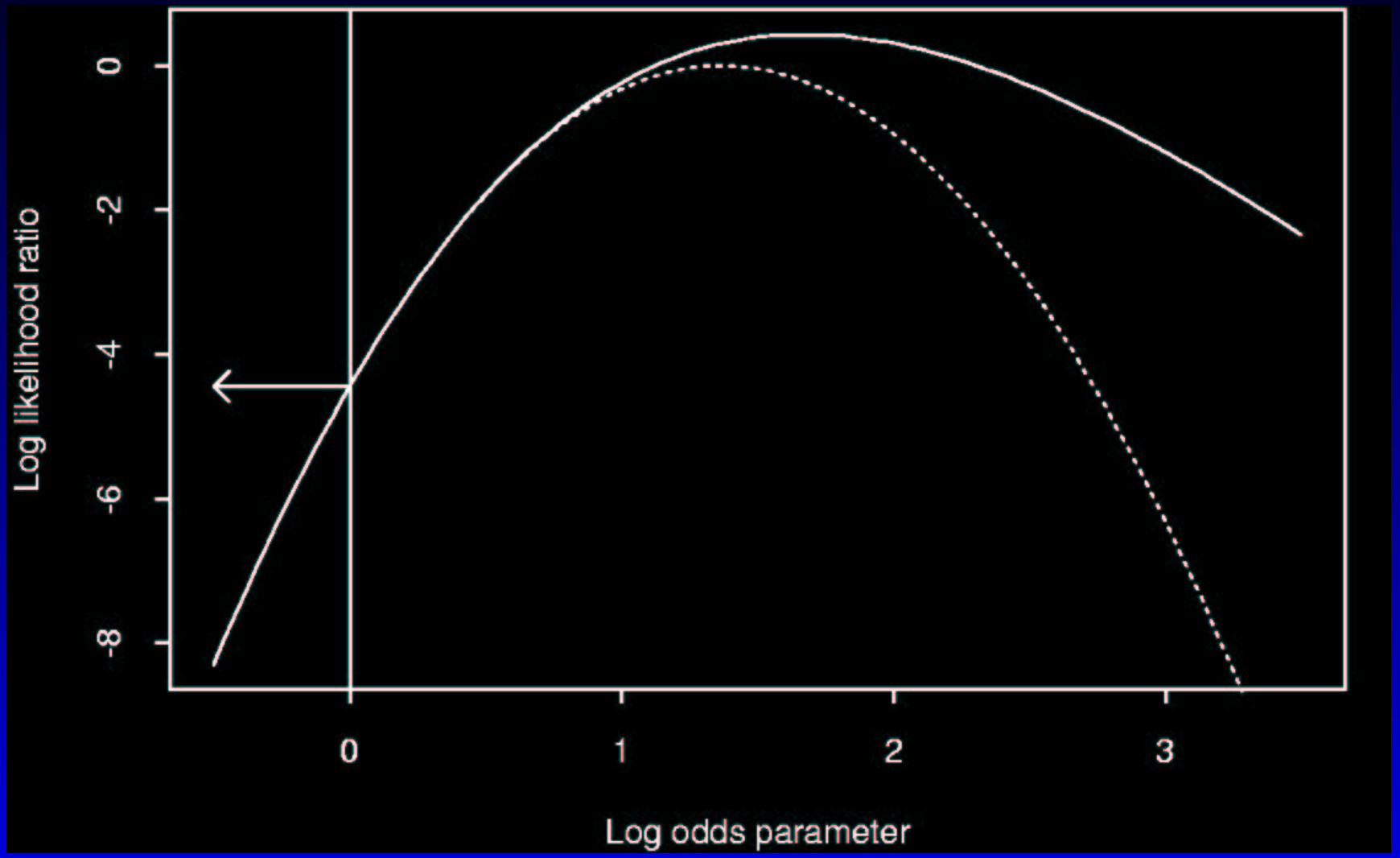
Likelihood ratio test



Wald test



Score test



Asymptotic properties

- The three tests are asymptotically equivalent
- Of the three, Wald tests have the worst small sample properties [a]
- Only two are implemented in a general way in *Stata* :
  - `lrtest` implements likelihood ratio test
  - `test, testparm` implement Wald tests

[a] Fears et al., *Am. Statitician*, **50:**227–7 (1996)

E.g. χ^2 tests in 2–way tables

- O and E represent observed and "expected" frequencies in cells of the table
- LR test: $X^2 = 2 \sum O \ln \frac{O}{E}$
- Score test: $X^2 = \sum \frac{(O-E)^2}{E}$
- Wald test: $X^2 = \sum \frac{(O-E)^2}{O}$
  (This assume the identity "link" function. The test is different for the more usual logit link)
- Note that the Wald test breaks down in the presence of empty cells

Robust variance estimates

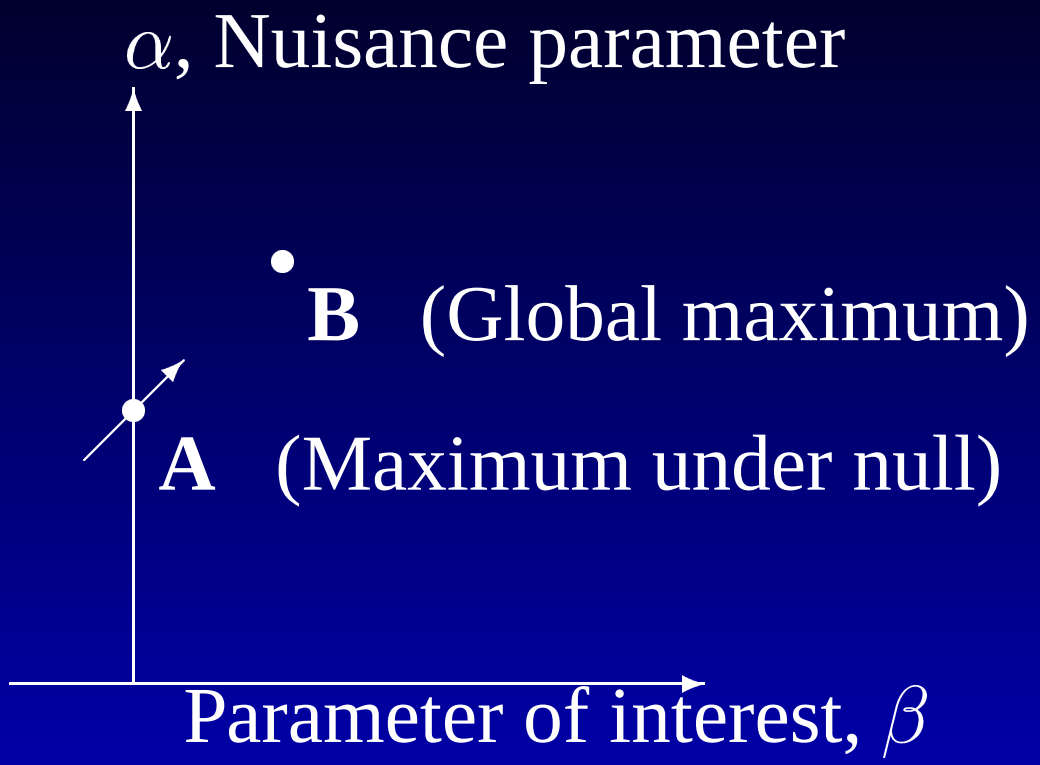
- Asymptotics depend on the fact that, *if the model is correct*, the information matrix estimates the variance of the score
- If variance is misspecified and/or data are "clustered", the LR test will be wrong
- The Wald test can be generalized by use of Huber–White variance estimate — estimate the variance of the score from the empirical variance of score contributions from independent clusters
- The same trick can be used for score tests [a] but is less well-known, and rarely implemented

[a] Boos, *Am. Statistician*, **46:**327–33 (1992)

Generalized score test

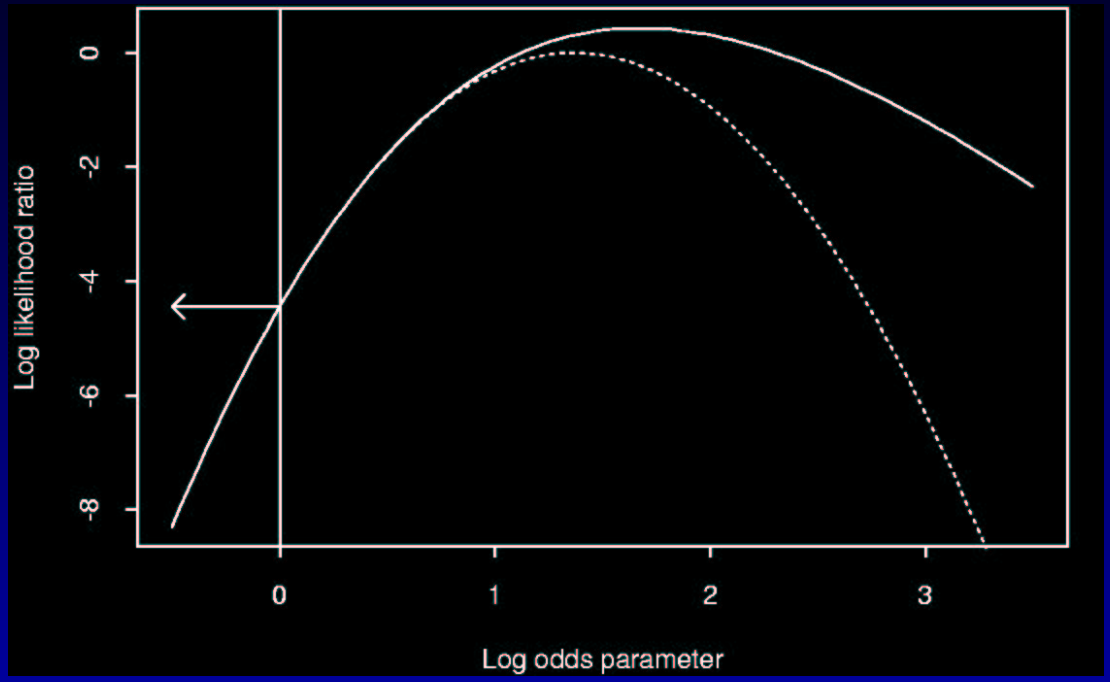
- In the absence of "nuisance parameters", the idea is easy:
  - Score vector, U, is the sum of independent contributions u_i. If its variance is V we calculate $X^2 = U^{\mathrm{T}} V^{\ominus} U$
  - Model-based test: estimate V from second derivative of log–likelihood
  - "Robust" test: estimates V by $\sum_i u_i u_i^{\mathrm{T}}$
- Things are more difficult in the presence of nuisance parameters, since test statistic depends on nuisance parameters and these have been chosen to maximize the (possibly incorrect) likelihood

A geometrical interpretation



- Test gradient in the direction **AB** — which depends on second derivative matrix at **A**
- Estimate variance empirically

Score test as 1-step estimate



- We can view the score test as the first step of a Newton-Raphson iteration
- This also applies with nuisance parameters, and can be generalized to the `robust` case

Computation

1. Maximize likelihood w.r.t. nuisance parameters, α — *i.e.*find point **A**
2. Evaluate full score vector and information matrix w.r.t. (α, β)
3. Calculate robust estimate of variance of score vector at **A** and hence the conventional "sandwich" variance estimator
4. Compute 1-step (Newton-Raphson) approximation to global ML estimate $(\widehat{\alpha}, \widehat{\beta})$
5. Conventional calculations for Wald test now yield the generalized score test

Required *Stata* implementation

1. We need a program, `sctest` say, which is called after an e-class command, and tests for adding new variables into the model:

```
. logit chd quetelet
. sctest bp, robust cluster(fmly)
```

2. `sctest` needs to work in conjunction with `xi:`

```
. logit chd quetelet bp
. xi:sctest i.grp, robust cluster(fmly)
```

3. Ideally it should work with all e-class commands

Our attempt

1. Evaluate subset of records which will be included in the full model
2. Refit model in restricted subset of records
3. Use `predict ,xb` to find linear predictor values at **A**
4. Refit *full* model
   - adding new variables,
   - with previously computed linear predictors as "offsets", and
   - specifying `iter(0)` to force 1-step
5. `testparm` the new parameters

Why doesn't it work?

- This relies on certain behaviour of e-class programs under the `iter(0)` option:
  1. After computing 1-step estimate of parameters, do not update second derivatives
  2. Likewise, do not recompute score contributions and the "meat" matrix of the information sandwich
- Neither `logit` or `poisson` behave like this. Nor do they behave in the same way as one another!

Coding the main iteration loop

This idea would work if, in all programs, the iteration loop were coded as follows:

```
while (it < iter & not converged) {
  Compute first derivative contributions
  Compute first and second derivatives
  (Optionally) calculate sandwich estimate
  Newton-Raphson update of  ML estimate
}
Print results and return
```

In fact, the various *Stata* commands seem to carry out some further updates before returning.

But it may still be useful

- Asymptotically the additional updates should have no effect
- Nevertheless, in view of these difficulties, the program is not on our web site, but is available to anyone interested
- The command:

  sctest *varlist* **[, robust cluster(***varname***)]**

An example

	$Y=1$	$Y=0$
$X=1$	7	3
$X=0$	3	7

Test	X^2
Score	3.20
Wald (identity link)	3.81
Wald (logit link)	3.02
`sctest`	2.74
`sctest, robust`	2.65