Adaptive dose-finding designs to identify multiple doses that achieve multiple response targets

Adrian Mander and Simon Bond

MRC Biostatistics Unit Hub for Trials Methodology Research

13 Sep 2013

MRC | Hubs for Trials Methodology Research

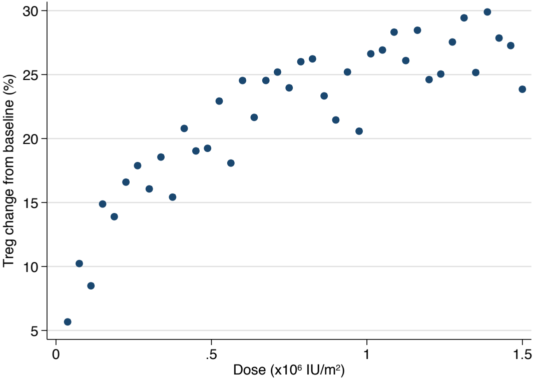
MRC Biostatistics Unit Hub

Motivating example - Diabetes IL-2 trial

- Immune response primary endpoint - (max) Treg % change from baseline over 5 days
- Injecting drug — any concentration is available
- One possible dose-response model is the non-linear Emax model

$$y = \theta_0 + \frac{\theta_1}{\theta_2 + \text{dose}}$$

$$y = f(dose, \theta)$$



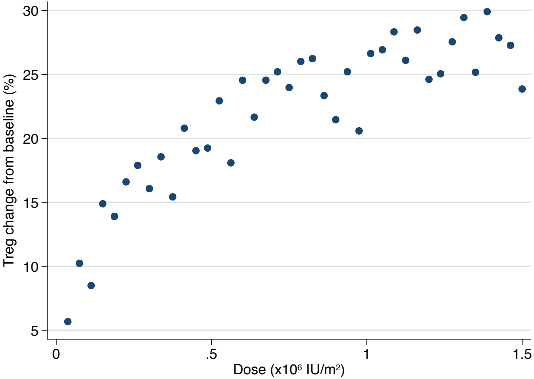
Either pick design to estimate the whole curve or targeted responses

Motivating example - Diabetes IL-2 trial

- Immune response primary endpoint - (max) Treg % change from baseline over 5 days
- Injecting drug — any concentration is available
- One possible dose-response model is the non-linear Emax model

$$y = \theta_0 + \frac{\theta_1}{\theta_2 + \text{dose}}$$

$$y = f(dose, \theta)$$



Either pick design to estimate the whole curve or targeted responses

D-optimal designs

A design (ξ) gives the doses (d_i) to choose and the relative frequency of patients on each dose (w_i)

$$\xi = \begin{Bmatrix} d_1 & d_2 & d_3 \\ w_1 & w_2 & w_3 \end{Bmatrix}$$

The **information matrix** for the design is defined as

$$M(\xi, \theta) = \sum_{i=1}^{3} w_i \frac{\partial f(d_i, \theta)}{\partial \theta} \frac{\partial f^T(d_i, \theta)}{\partial \theta}$$

The **D-optimal design** maximises the (log) determinant of the information matrix:

$$\arg\max_{\xi} |M(\xi, \theta)|$$

D-optimal designs

A design (ξ) gives the doses (d_i) to choose and the relative frequency of patients on each dose (w_i)

$$\xi = \begin{Bmatrix} d_1 & d_2 & d_3 \\ w_1 & w_2 & w_3 \end{Bmatrix}$$

The **information matrix** for the design is defined as

$$M(\xi, \theta) = \sum_{i=1}^{3} w_i \frac{\partial f(d_i, \theta)}{\partial \theta} \frac{\partial f^T(d_i, \theta)}{\partial \theta}$$

The **D-optimal design** maximises the (log) determinant of the information matrix:

$$\arg\max_{\xi} |M(\xi, \theta)|$$

doptimal.ado

Either given θ what is the D-optimal design OR estimate θ from a dataset to give the **locally D-optimal** design.

```
doptimal, theta(0 30 0.2) model(emax) mindose(0) maxdose(1.5)
doptimal using temp.dta,  model(emax) mindose(0) maxdose(1.5)
```

Output is

```
Estimating model parameters for Emax model given the dataset
Finding D-optimal design

The model parameters for the emax model are
 -3.7108234 32.536035 .13706979

The D-optimal design is

  Doses   9.42e-175    .1159005     1.5
Weights   .33454905    .33090189    .33454905
```

doptimal.ado

Either given θ what is the D-optimal design OR estimate θ from a dataset to give the **locally D-optimal** design.

```
doptimal, theta(0 30 0.2) model(emax) mindose(0) maxdose(1.5)
doptimal using temp.dta,  model(emax) mindose(0) maxdose(1.5)
```

Output is

```
Estimating model parameters for Emax model given the dataset
Finding D-optimal design

The model parameters for the emax model are
 -3.7108234 32.536035 .13706979

The D-optimal design is

  Doses  9.42e-175   .1159005    1.5
Weights  .33454905   .33090189   .33454905
```

A few thoughts

- The models available in o͡ptimal.ado are linear, quadratic and 4-parameter logistic
- Optimize needs to optimize over a vector rather than a matrix
- Constraints i.e. weights sum to 1 and weights are between 0 and 1 doses are between mindose maxdose
- I plan to adapt the code to accept a vector of differentials for user-specified functions
- Symbolic differentiation in R

A few thoughts

- The models available in ȏptimal.ado are linear, quadratic and 4-parameter logistic
- Optimize needs to optimize over a vector rather than a matrix
- Constraints i.e. weights sum to 1 and weights are between 0 and 1 doses are between mindose maxdose
- I plan to adapt the code to accept a vector of differentials for user-specified functions
- Symbolic differentiation in R

A few thoughts

- The models available in ȏptimal.ado are linear, quadratic and 4-parameter logistic
- Optimize needs to optimize over a vector rather than a matrix
- Constraints i.e. weights sum to 1 and weights are between 0 and 1 doses are between mindose maxdose
- I plan to adapt the code to accept a vector of differentials for user-specified functions
- Symbolic differentiation in R

Il-2 study

The study team wanted to estimate the dose-response curve but then decided to target 10% and 20% responses

- need to use the inverse function $f^{-1}(y, \theta) = d$
- e.g. for 20% response the dose $d_{0.2} = f^{-1}(0.2, \theta) = \mathbf{g(0.2}, \theta)$.

We designed a study for 10 initial patients

- pairs of patients were put on the doses 0.04, 0.16, 0.6, 1 and 1.5 $(IU x 10^6 / m^2)$

Then we select the "best" model to determine future doses to minimize variances of targeted doses.

Il-2 study

The study team wanted to estimate the dose-response curve but then decided to target 10% and 20% responses

- need to use the inverse function $f^{-1}(y, \theta) = d$
- e.g. for 20% response the dose $d_{0.2} = f^{-1}(0.2, \theta) = \mathbf{g}(\mathbf{0.2}, \theta)$.

We designed a study for 10 initial patients

- pairs of patients were put on the doses 0.04, 0.16, 0.6, 1 and 1.5 $(IUx10^6/m^2)$

Then we select the "best" model to determine future doses to minimize variances of targeted doses.

Generic method for a single target response

Need to calculate the (expected) **information matrix** after k patients for the design is defined as

$$\begin{aligned} M_k(\theta) &= \sum_{i=1}^{k} \frac{\partial f(d_i,\theta)}{\partial \theta} \frac{\partial f^T(d_i,\theta)}{\partial \theta} \\ Var_k(\theta) &\approx \sigma_e^2 M_k^{-1}(\theta) \end{aligned}$$

To select the next dose d^* for patient $k+1$ that minimises $Var_{k+1}(d_{0.2},\theta)$

Generic method for a single target response

Need to calculate the (expected) **information matrix** after k patients for the design is defined as

$$\begin{aligned} M_k(\theta) &= \sum_{i=1}^{k} \frac{\partial f(d_i,\theta)}{\partial\theta}\frac{\partial f^T(d_i,\theta)}{\partial\theta} \\ Var_k(\theta) &\approx \sigma_e^2 M_k^{-1}(\theta) \end{aligned}$$

To select the next dose d^* for patient $k+1$ that minimises $Var_{k+1}(d_{0.2},\theta)$

The algorithm

$$Var_k(\theta) = \quad \sigma_e^2 M_k^{-1}(\theta)$$

Using the delta method

$$Var_k(d_{0.2}|\theta) \approx \quad \frac{\partial g^T(0.2,\theta)}{\partial\theta} Var_k(\theta) \frac{\partial g(0.2,\theta)}{\partial\theta} \tag{1}$$

Now need to pick d^* and recalculate the above two equations

$$Var^*_{k+1}(\theta) = \sigma_e^2 \left(M_k(\theta) + \frac{\partial f(d^*,\theta)}{\partial\theta} \frac{\partial f^T(d^*,\theta)}{\partial\theta} \right)^{-1}$$

- Set θ to be $\hat{\theta}^{(k)}$, the estimate after k patients.
- Then plug $Var_{k+1}(d_{0.2}|\hat{\theta}^{(k)}, d^*)$ into equation (1)
- use `optimize` to minimise this function wrt d^*

The algorithm

$$Var_k(\theta) = \quad \sigma_e^2 M_k^{-1}(\theta)$$

Using the delta method

$$Var_k(d_{0.2}|\theta) \approx \quad \tfrac{\partial g^T(0.2,\theta)}{\partial\theta} Var_k(\theta) \tfrac{\partial g(0.2,\theta)}{\partial\theta} \tag{1}$$

Now need to pick d^* and recalculate the above two equations

$$Var^*_{k+1}(\theta) = \sigma_e^2 \left(M_k(\theta) + \frac{\partial f(d^*,\theta)}{\partial\theta} \frac{\partial f^T(d^*,\theta)}{\partial\theta} \right)^{-1}$$

- Set θ to be $\hat{\theta}^{(k)}$, the estimate after k patients.
- Then plug $Var_{k+1}(d_{0.2}|\hat{\theta}^{(k)}, d^*)$ into equation (1)
- use `optimize` to minimise this function wrt d^*

The algorithm

$$Var_k(\theta) = \quad \sigma_e^2 M_k^{-1}(\theta)$$

Using the delta method

$$Var_k(d_{0.2}|\theta) \approx \quad \frac{\partial g^T(0.2,\theta)}{\partial\theta} Var_k(\theta) \frac{\partial g(0.2,\theta)}{\partial\theta} \tag{1}$$

Now need to pick d^* and recalculate the above two equations

$$Var^*_{k+1}(\theta) = \sigma_e^2 \left(M_k(\theta) + \frac{\partial f(d^*,\theta)}{\partial\theta} \frac{\partial f^T(d^*,\theta)}{\partial\theta} \right)^{-1}$$

- Set θ to be $\hat{\theta}^{(k)}$, the estimate after k patients.
- Then plug $Var_{k+1}(d_{0.2}|\hat{\theta}^{(k)}, d^*)$ into equation (1)
- use `optimize` to minimise this function wrt d^*

The algorithm

$$Var_k(\theta) = \quad \sigma_e^2 M_k^{-1}(\theta)$$

Using the delta method

$$Var_k(d_{0.2}|\theta) \approx \quad \tfrac{\partial g^T(0.2,\theta)}{\partial \theta} Var_k(\theta) \tfrac{\partial g(0.2,\theta)}{\partial \theta} \tag{1}$$

Now need to pick d^* and recalculate the above two equations

$$Var^*_{k+1}(\theta) = \sigma_e^2 \left(M_k(\theta) + \frac{\partial f(d^*, \theta)}{\partial \theta} \frac{\partial f^T(d^*, \theta)}{\partial \theta} \right)^{-1}$$

- Set θ to be $\hat{\theta}^{(k)}$, the estimate after k patients.
- Then plug $Var_{k+1}(d_{0.2}|\hat{\theta}^{(k)}, d^*)$ into equation (1)
- use `optimize` to minimise this function wrt d^*

core of my Mata code

```
Sbeta =optimize_init()
optimize_init_evaluator(Sbeta, &findbetahat_emax())
optimize_init_which(Sbeta, "min")
optimize_init_params(Sbeta, (0,1.25,1) )
optimize_init_argument(Sbeta, 1, data)
betahat=optimize(Sbeta)

e = (data[,2]-f_emax(betahat,data[,1]))
resvarhat = e'e/(rows(data)-3)
OIk = OIk_emax(betahat, data, target)
```

```
S=optimize_init()
optimize_init_evaluator(S, &findD_emax())
optimize_init_which(S, "min")
optimize_init_params(S, egbeta_emax(betahat, target))
optimize_init_argument(S, 1, OIk)
optimize_init_argument(S, 2, betahat)
optimize_init_argument(S, 3, resvarhat)
optimize_init_argument(S, 4, target)
p=optimize(S)
```

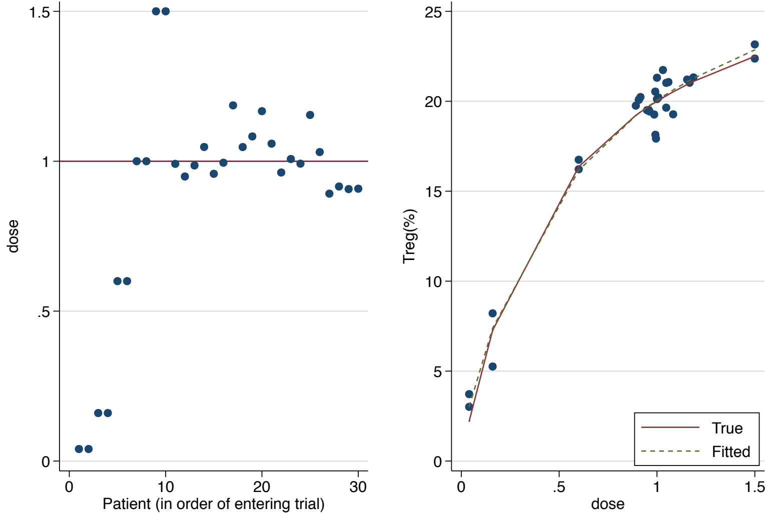
core of my Mata code

```
Sbeta =optimize_init()
optimize_init_evaluator(Sbeta, &findbetahat_emax())
optimize_init_which(Sbeta, "min")
optimize_init_params(Sbeta, (0,1.25,1) )
optimize_init_argument(Sbeta, 1, data)
betahat=optimize(Sbeta)

e = (data[,2]-f_emax(betahat,data[,1]))
resvarhat = e'e/(rows(data)-3)
OIk = OIk_emax(betahat, data, target)


S=optimize_init()
optimize_init_evaluator(S, &findD_emax())
optimize_init_which(S, "min")
optimize_init_params(S, egbeta_emax(betahat, target))
optimize_init_argument(S, 1, OIk)
optimize_init_argument(S, 2, betahat)
optimize_init_argument(S, 3, resvarhat)
optimize_init_argument(S, 4, target)
p=optimize(S)
```

A single simulation



Res var= 1

Two targets

To handle the teams desire for two targeted doses we need the variance-covariance matrix of $(d_{0.1}, d_{0.2})$, we can minimise either

- The trace $Var(d_{0.1}, d_{0.2})$, or
- Determinant of $Var(d_{0.1}, d_{0.2})$.

Both performed well BUT one feature of the trial design that the investigators desired was dosing patients close to the two targets

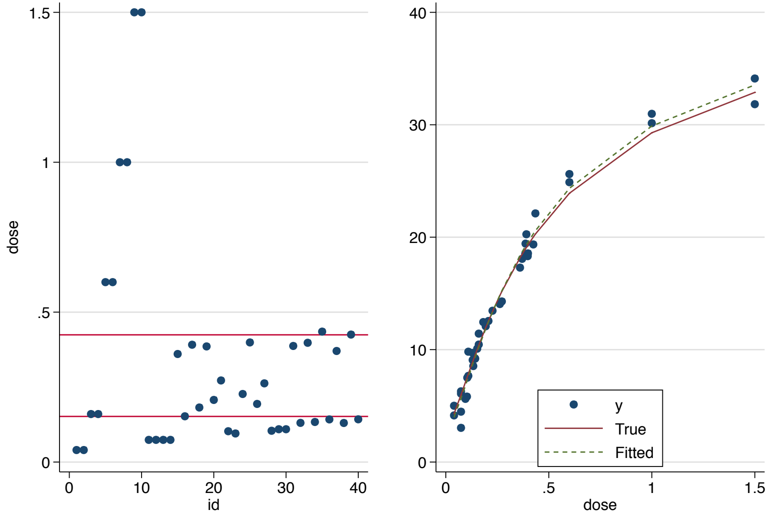
- models with fewer than 4 parameters suggested doses in the middle of the two targeted doses.

Algorithm outline

Want to use optimise to find doses d_1^*, d_2^*, i.e. we want to find the next 2 doses.

- Need to add the two extra bits of information into $Var_{k+2}(\theta)$
- then calculate the trace using this variance estimate
  - $Var(d_{0.1}) + Var(d_{0.2})$

A single simulation



Res var= 1

Current state of the trial

The trial had a dose-decision meeting yesterday after the first 10 patients. I never asked permission to display the data but the non-linear models are fitting OK and the variance of the response is low.

Acknowledements

I would like to thank Frank Waldron-Lynch, John Todd, Linda Wicker and Larry (CIMR) for involving us in their study.

Current state of the trial

The trial had a dose-decision meeting yesterday after the first 10 patients. I never asked permission to display the data but the non-linear models are fitting OK and the variance of the response is low.

Acknowledements

I would like to thank Frank Waldron-Lynch, John Todd, Linda Wicker and Larry (CIMR) for involving us in their study.

Conclusions

I have produced two Stata commands

- `optimal.ado` — to give the D-optimal design
- `il2.ado` — more bespoke dose-finding function that I hope to make more generic (if interest)
  - these methods are still evolving.
  - the numerical methods of optimize() and deriv() seem to be holding up to this command