Optimising Sequential and Adaptive Designs:

The Power of Dynamic Programming

Christopher Jennison

Department of Mathematical Sciences,

University of Bath, UK

http://people.bath.ac.uk/mascj

Adaptive Designs and Multiple Testing Procedures Workshop

Basel

April 2023

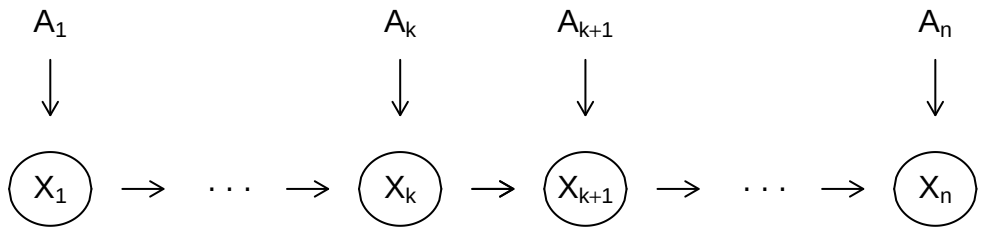
Outline of talk

1. Dynamic Programming
   - General description
2. Optimising a group sequential stopping rule
   - Problem formulation and optimisation
   - What we learn from this
   - Related problems
3. Optimising dose allocation in a First in Human trial
   - Problem formulation and optimisation
   - Optimising a trial with safety and efficacy endpoints
   - Some computational short cuts

1. Dynamic Programming (Bellman, *PNAS*, 1953)

Dynamic Programming is a computationally efficient method for solving a sequential decision problem.

In a problem with n stages, denote the "State" in stage k by X_k and the "Action" applied by A_k.



State X_{k+1} depends on X_k and A_k, and may be random.

The aim is to choose the actions $A_1, \ldots, A_n$ so as to maximise the expected value of a gain or utility function

$$U(X_1, A_1, \ldots, X_n, A_n),$$

or to minimise the expected value of a loss function

$$L(X_1, A_1, \ldots, X_n, A_n).$$

Dynamic Programming

I first met Dynamic Programming in an undergraduate course.

The problems we were able to solve were discrete, for example, the "car parking" problem:

> *We start a distance N parking places from our destination. As we cruise along, we can see only one parking place at a time. If a place is empty and we park there, then our loss is the distance we walk. Clearly, once past our destination we park in the first available place. Empty spaces occur independently with probability $p = 1 - q$. Show that the optimal policy is to wait until we are N_0 places from our destination and then park in the first available space, where N_0 is the largest integer $r > 0$ with $q^r \geq 1/2$.*

Problems with a continuous state space did not have "nice" solutions and we were not shown the numerical methods that might be applied to solve them.

Dynamic Programming

Dynamic Programming works backwards from the final stage.

Suppose we are minimising a loss function $L(X_1, A_1, \ldots, X_n, A_n)$.

At stage n

We consider each possibility for the current state X_n and the preceding history $X_1, A_1, \ldots, X_{n-1}, A_{n-1}$.

We determine the action A_n^* that minimises the conditional expected loss given $X_1, A_1, \ldots, X_n$ and we define

$$\beta^{(n)}(X_1, A_1, \ldots, X_n)$$

to be this conditional expected loss for the optimal action, A_n^*.

We store the optimal action A_n^* and conditional expected loss $\beta^{(n)}(X_1, A_1, \ldots, X_n)$ for every possible sequence $X_1, A_1, \ldots, X_n$.

Dynamic Programming

At stage $n-1$

For each state X_{n-1} and previous history $X_1, A_1, \ldots, X_{n-2}, A_{n-2}$, we determine the action A^*_{n-1} that minimises the conditional expected loss, assuming we shall act optimally at stage n.

In doing this, we use the function $\beta^{(n)}(X_1, A_1, \ldots, X_n)$ that we calculated previously.

We define

$$\beta^{(n-1)}(X_1, A_1, \ldots, X_{n-1})$$

to be the conditional expected loss when taking the action A^*_{n-1} and continuing to proceed optimally thereafter.

We store the optimal action A^*_{n-1} and conditional expected loss $\beta^{(n-1)}(X_1, A_1, \ldots, X_{n-1})$ for every sequence $X_1, A_1, \ldots, X_{n-1}$.

Dynamic Programming

Continuing backwards . . .

We find the optimal actions for each possible state at stage $n-2$, stage $n-3$, . . . , stage 1.

At each stage k

We use the function $\beta^{(k+1)}(X_1, A_1, \ldots, X_{k+1})$ that we calculated previously and store values of the function $\beta^{(k)}(X_1, A_1, \ldots, X_k)$ for future use.

Carrying out the computations

The complexity of the above calculations depends on the nature of the states $X_1, \ldots, X_n$, the process that determines successive states, and the form of the loss function being minimised.

In the following examples we shall see it can be surprisingly easy to solve difficult problems — and there are ways to tackle apparently impossible problems.

2. Optimising a group sequential clinical trial

Reference: Jennison & Turnbull, *Kuwait Journal of Science*, 2013.

Consider a Phase 3 clinical trial comparing a new treatment against a standard.

Let θ denote the "effect size", a measure of the improvement in the new treatment over the standard.

We shall test the null hypothesis H_0: $\theta \leq 0$ against $\theta > 0$.

Rejecting H_0 allows us to conclude the new treatment is superior.

We allow type I error probability α for rejecting H_0 when it is true.

We specify power $1 - \beta$ as the probability of rejecting H_0 when $\theta = \delta$. Here δ is, typically, the minimal clinically significant treatment difference.

The trial design, including the method of analysis and stopping rule, must be set up to attain these error rates.

Sequential distribution theory

Let $\widehat{\theta}_k$ denote the estimate of the treatment effect θ at analysis k.

Information for θ at analysis k is $\mathcal{I}_k = \{\text{Var}(\widehat{\theta}_k)\}^{-1},\ k = 1, \ldots, K.$

Canonical joint distribution of $\widehat{\theta}_1, \ldots, \widehat{\theta}_K$

In many situations, in the absence of early stopping, $\widehat{\theta}_1, \ldots, \widehat{\theta}_K$ are approximately multivariate normal,

$$\widehat{\theta}_k \sim N(\theta, \{\mathcal{I}_k\}^{-1}), \quad k = 1, \ldots, K,$$

and

$$\text{Cov}(\widehat{\theta}_{k_1}, \widehat{\theta}_{k_2}) = \text{Var}(\widehat{\theta}_{k_2}) = \{\mathcal{I}_{k_2}\}^{-1} \quad \text{for } k_1 < k_2.$$

References:

Jennison & Turnbull, *JASA*, 1997,

Scharfstein et al, *JASA*, 1997.

An optimal stopping problem

Consider a trial designed to test H_0: $\theta \leq 0$ vs $\theta > 0$, with:

Type I error rate α,

Power $1 - \beta$ at $\theta = \delta$,

Up to K analyses.

A fixed sample test needs information

$$\mathcal{I}_{fix} = \{\Phi^{-1}(\alpha) + \Phi^{-1}(\beta)\}^2/\delta^2.$$

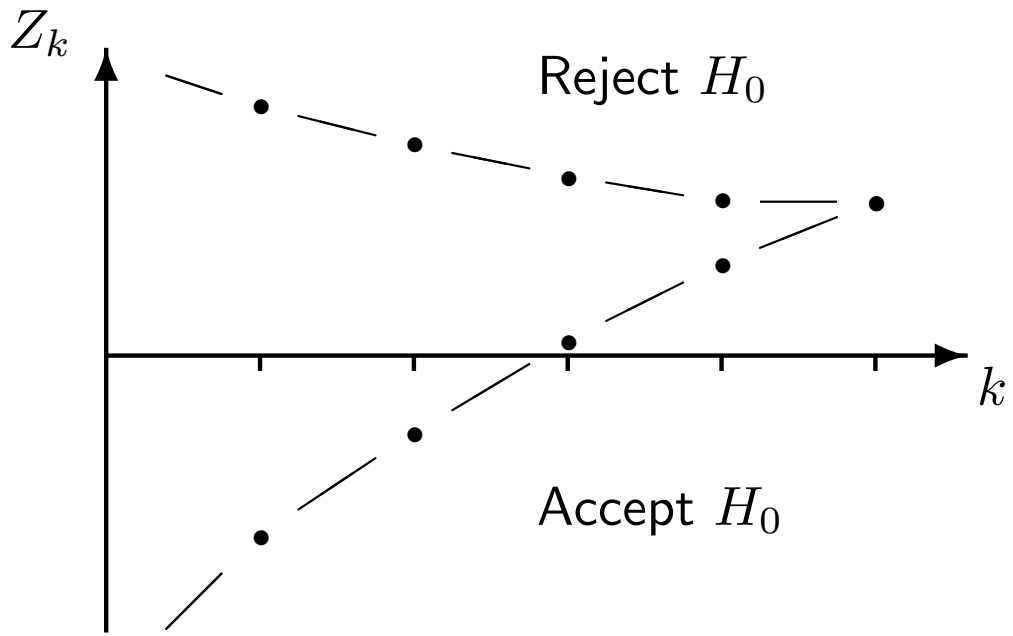
We set the maximum information to be

$$\mathcal{I}_{max} = R\mathcal{I}_{fix},$$

where $R > 1$, with equal increments between analyses.

Optimal group sequential tests

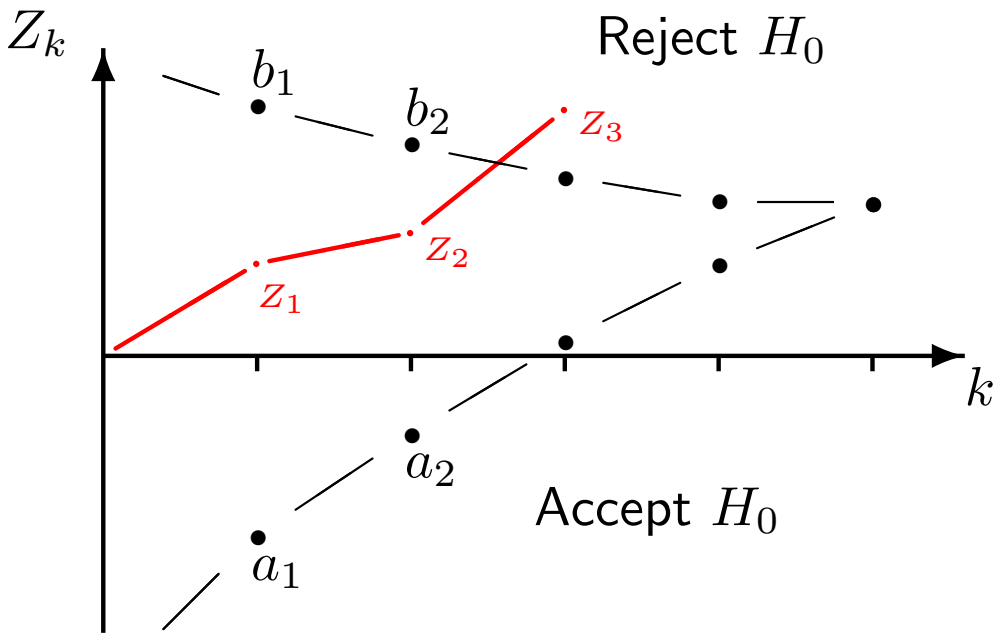
The error rates impose two constraints on the $2K-1$ boundary points — leaving a high dimensional space of possible boundaries.



We shall look for a boundary that minimises

$$\{E_0(\mathcal{I}) + E_\delta(\mathcal{I})\}/2.$$

Computations for group sequential tests



We need to be able to calculate the probabilities of basic events such as

$$a_1 < Z_1 < b_1, \quad a_2 < Z_2 < b_2, \quad Z_3 > b_3.$$

Combining such probabilities gives key properties, such as $Pr_\theta\{\text{Reject } H_0\}$ and $E_\theta(\mathcal{I})$.

Numerical integration

We can write probabilities as nested integrals, e.g.,

$$Pr\{a_1 < Z_1 < b_1,\, a_2 < Z_2 < b_2,\, Z_3 > b_3\} =$$

$$\int_{a_1}^{b_1} \int_{a_2}^{b_2} \int_{b_3}^{\infty} f_1(z_1)\, f_2(z_2|z_1)\, f_3(z_3|z_2)\, dz_3\, dz_2\, dz_1.$$

Applying numerical integration, we replace each integral by a sum of the form

$$\int_a^b f(z)\, dz \;=\; \sum_{i=1}^{n} w(i)\, f(z(i)),$$

where $z(1), \ldots, z(n)$ is a grid of points from a to b.

Numerical integration

Thus, we have

$$Pr\{a_1 < Z_1 < b_1,\, a_2 < Z_2 < b_2,\, Z_3 > b_3\} \approx$$

$$\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \sum_{i_3=1}^{n_3} w_1(i_1)\, f_1(z_1(i_1))\; w_2(i_2) f_2(z_2(i_2)|z_1(i_1))$$

$$w_3(i_3)\; f_3(z_3(i_3)|z_2(i_2)).$$

Multiple integrations and summations will arise, e.g., for an outcome at analysis k,

$$\sum_{i_1=1}^{n_1} \ldots \sum_{i_k=1}^{n_k} w_1(i_1)\, f_1(z_1(i_1))\; w_2(i_2) f_2(z_2(i_2)|z_1(i_1))$$

$$\ldots\; w_k(i_k)\; f_k(z_k(i_k)|z_{k-1}(i_{k-1})).$$

Numerical integration

In the multiple summation

$$\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_k=1}^{n_k} w_1(i_1)\, f_1(z_1(i_1))\; w_2(i_2) f_2(z_2(i_2)|z_1(i_1))$$

$$\cdots\; w_k(i_k)\; f_k(z_k(i_k)|z_{k-1}(i_{k-1})),$$

the structure of the k nested summations is such that the computation required is of the order of $k-1$ double summations.

Using Simpson's rule with 100 to 200 grid points per integral can give accuracy to 5 or 6 decimal places.

For details of efficient sets of grid points, see Ch. 19 of *Group Sequential Methods with Applications to Clinical Trials* by Jennison and Turnbull (2000).

Finding optimal group sequential tests

Recall, we want a group sequential test of H_0: $\theta \leq 0$ vs $\theta > 0$ with

$$Pr_{\theta=0}\{\text{Reject } H_0\} = \alpha,$$

$$Pr_{\theta=\delta}\{\text{Accept } H_0\} = \beta,$$

Analyses at $\mathcal{I}_k = (k/K)\,\mathcal{I}_{max}, \quad k = 1, \ldots, K,$

Minimum possible value of $\{E_0(\mathcal{I}) + E_\delta(\mathcal{I})\}/2.$

We deal with constraints on error rates by introducing Lagrangian multipliers to create the *unconstrained problem* of minimising

$$\{E_0(\mathcal{I}) + E_\delta(\mathcal{I})\}/2 + \lambda_1 Pr_{\theta=0}\{\text{Reject } H_0\} + \lambda_2\, Pr_{\theta=\delta}\{\text{Accept } H_0\}.$$

We shall find a pair of multipliers (λ_1, λ_2) such that the solution has type I and II error rates α and β, then this design will solve the *constrained problem* too.

Bayesian interpretation of the Lagrangian approach

Suppose we put a prior on θ with $Pr\{\theta = 0\} = Pr\{\theta = \delta\} = 0.5$ and specify costs of

$$
\begin{array}{ll}
1 & \text{per unit of information observed,} \\
2\,\lambda_1 & \text{for rejecting } H_0 \text{ when } \theta = 0, \\
2\,\lambda_2 & \text{for accepting } H_0 \text{ when } \theta = \delta.
\end{array}
$$

Then, the total Bayes risk is

$$\{E_0(\mathcal{I}) + E_\delta(\mathcal{I})\}/2 + \lambda_1\, Pr_{\theta=0}\{\text{Reject } H_0\} + \lambda_2\, Pr_{\theta=\delta}\{\text{Accept } H_0\},$$

just as in the Lagrangian problem.

An advantage of the Bayes interpretation is that it can give insight into solving the problem by using "Dynamic Programming" or "Backwards Induction".

Solution by Dynamic Programming

Denote the posterior distribution of θ given $Z_k = z_k$ at analysis k by

$$p^{(k)}(\theta|z_k), \quad \theta = 0, \delta.$$

At the final analysis, K

There is no further sampling cost, so compare decisions

$$\text{Reject } H_0: \qquad E(\text{Cost}) = 2\,\lambda_1\,p^{(K)}(0|z_K),$$

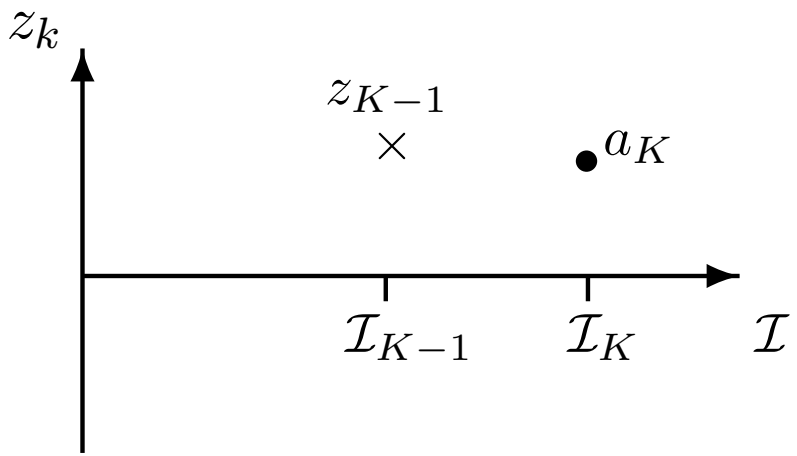
$$\text{Accept } H_0: \qquad E(\text{Cost}) = 2\,\lambda_2\,p^{(K)}(\delta|z_K).$$

The boundary point a_K is the value of z_K where these expected losses are equal.

The optimum decision rule is to reject H_0 for $Z_K > a_K$.

Dynamic Programming

At analysis $K-1$



If the trial stops at this analysis, there is no further cost of sampling and the expected additional cost is

$$\text{Reject } H_0: \qquad 2\,\lambda_1\,p^{(K-1)}(0|z_{K-1}),$$

$$\text{Accept } H_0: \qquad 2\,\lambda_2\,p^{(K-1)}(\delta|z_{K-1}).$$

At analysis $K-1$

If the trial continues to analysis K, the expected additional cost is

$$1 \times (\mathcal{I}_K - \mathcal{I}_{K-1})$$

$$+ 2\,\lambda_1\, p^{(K-1)}(0|z_{K-1})\, Pr_{\theta=0}\{Z_K > a_K | Z_{K-1} = z_{K-1}\}$$

$$+ 2\,\lambda_2\, p^{(K-1)}(\delta|z_{K-1})\, Pr_{\theta=\delta}\{Z_K < a_K | Z_{K-1} = z_{K-1}\}.$$

We can now define the optimal boundary points:

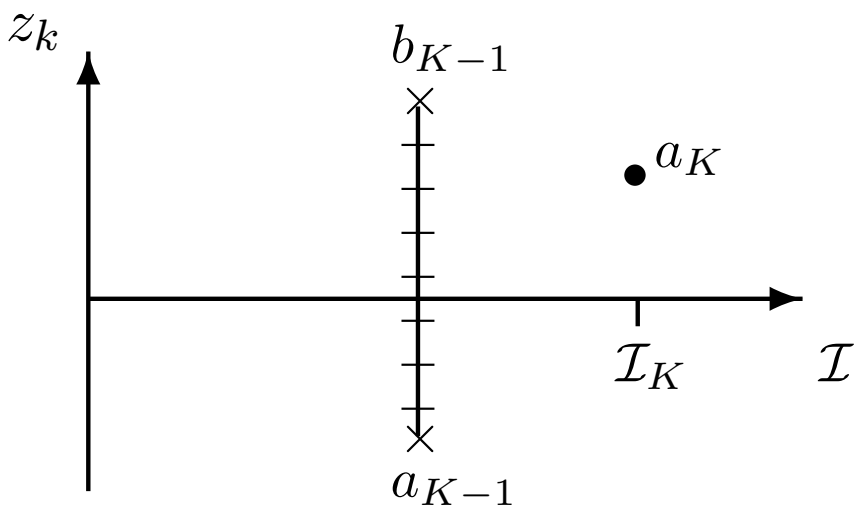
Set b_{K-1} to be the value of z_{K-1} where

$E(\text{Cost of continuing}) \;=\; E(\text{Cost of stopping to reject } H_0).$

Set a_{K-1} to be the value of z_{K-1} where

$E(\text{Cost of continuing}) \;=\; E(\text{Cost of stopping to accept } H_0).$

At analysis $K-1$



Before leaving analysis $K-1$, we set up a grid of points for use in numerical integration over the range a_{K-1} to b_{K-1}.

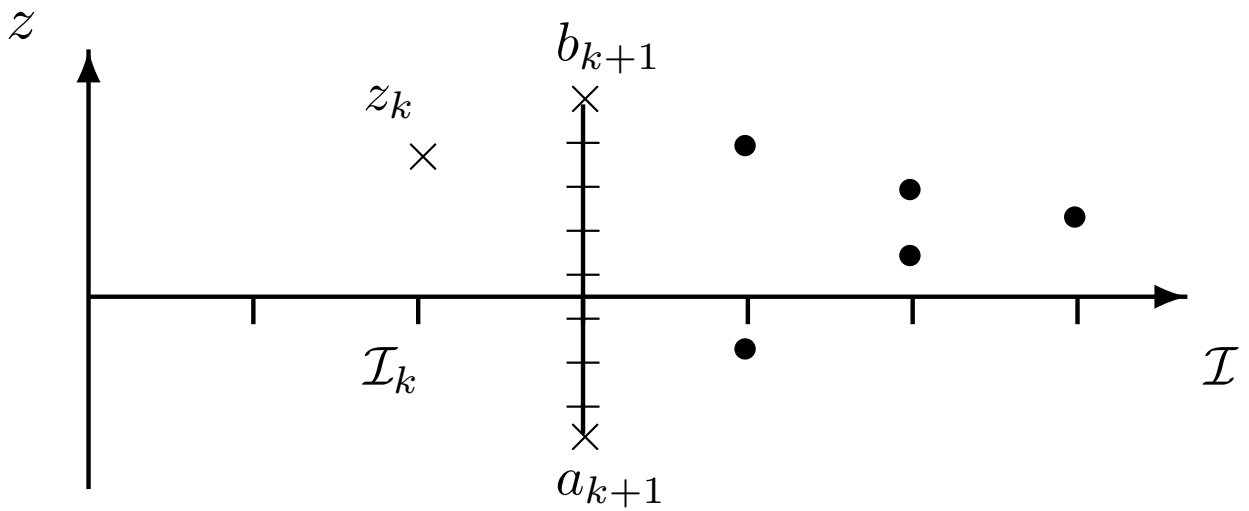
For each point, we sum over the posterior distribution of θ to calculate

$$\beta^{(K-1)}(z_{K-1}) = E(\text{Additional cost when continuing} \,|\, Z_{K-1} = z_{K-1}).$$

We are now ready to move back to analysis $K-2$.

Analyses 1 to $K-2$

We work back through analyses $k = K-2,\ K-3,\ \ldots,\ 1$.



At each analysis, we find the optimal stopping boundary using knowledge of the optimal stopping rule at future analyses.

Then, for a grid of values of z_k, compute

$$\beta^{(k)}(z_k) = E(\text{Additional cost when continuing} \,|\, Z_k = z_k)$$

to use in evaluating the option of continuing at analysis $k-1$.

Solving the original problem

For any given (λ_1, λ_2) we can find the Bayes optimal design and compute its type I and II error rates.

We now search for a pair (λ_1, λ_2) for which type I and type II error rates of the optimal design equal α and β, respectively.

The resulting design will be the optimal group sequential test, with the specified frequentist error rates, for our original problem.

Notes

1. The method of solving the overall problem demonstrates explicitly that good frequentist procedures should be similar to Bayes procedures.

2. The prior and costs in the final Bayes problem are a means to an end, rather than "true" costs of type I and type II errors, or costs of treating patients in the trial.

Properties of optimal designs

Tests with $\alpha = 0.025$, $1 - \beta = 0.9$, K analyses, $\mathcal{I}_{max} = R\mathcal{I}_{fix}$, and equal group sizes, that minimise $\{E_0(\mathcal{I}) + E_\delta(\mathcal{I})\}/2$.

Minimum values of $\{E_0(\mathcal{I}) + E_\delta(\mathcal{I})\}/2$, as a percentage of $\mathcal{I}_{fix}$

	R					
K	1.01	1.05	1.1	1.2	1.3	*Minimum over R*
2	80.8	74.7	73.2	73.7	75.8	73.0 at R=1.13
5	72.2	65.2	62.2	59.8	59.0	58.8 at R=1.38
10	69.2	62.2	59.0	56.3	55.1	54.2 at R=1.6
20	67.8	60.6	57.5	54.6	53.3	51.7 at R=1.8

Observe: $E(\mathcal{I}) \searrow$ as $K \nearrow$ but with diminishing returns,

$E(\mathcal{I}) \searrow$ as $R \nearrow$ up to a point.

Optimisation problems we have addressed

One-sided and two-sided group sequential tests

Eales & Jennison, *Biometrika*, 1992

Eales & Jennison, *Sequential Analysis*, 1995

Barber & Jennison, *Biometrika*, 2002

Group sequential tests with data dependent group sizes

Jennison & Turnbull, *Biometrika*, 2006

Group sequential tests of superiority and non-inferiority

Öhrn & Jennison, *Statistics in Medicine*, 2010

Group sequential tests for delayed responses

Hampson & Jennison, *JRSS, B*, 2013

Optimising gain functions from financial models

Robbie Peck, *University of Bath, PhD thesis*, 2020

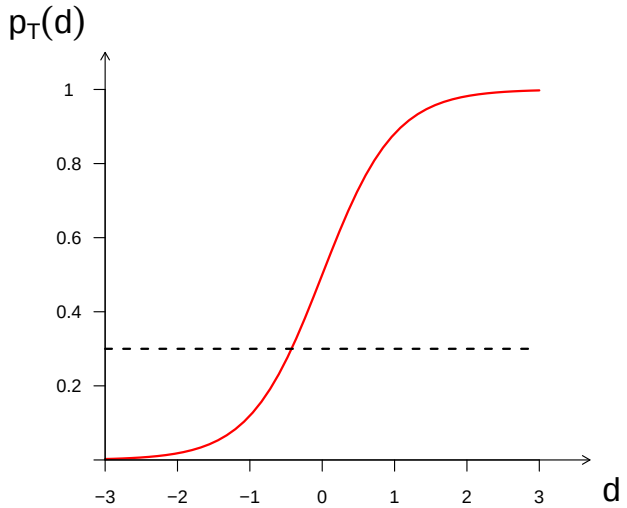
3. Optimising dose allocation in a First in Human trial

Reference: Lizzi Pitt, *University of Bath, PhD thesis*, 2021.

Phase I, First in Human, trials are conducted to investigate the safety of a new molecule and find the maximum tolerated dose.

Patients are treated a few at a time, e.g., in cohorts of 3.

The dose escalation scheme only moves on to a higher dose when lower doses have been shown to be safe.



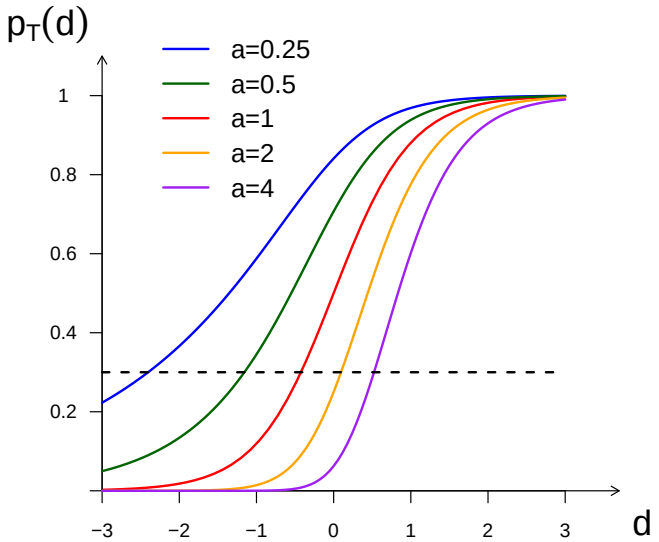
The aim is to determine the Maximum Tolerated Dose (MTD), at which the probability of a Dose Limiting Event (DLE) is, say, 0.3.

Dose response function (O'Quigley, et al. 1990, *Bmcs*)

We assume a dose response function of the form

$$p_T(d,a) = \left(\frac{\tanh(d)+1}{2}\right)^a = \left(\frac{1}{1+e^{-2d}}\right)^a,$$

where $p_T(d,a)$ is the probability of a DLE at dose level d, on some transformed scale (e.g., logarithmic).



The parameter a determines the likelihood of a DLE.

As a increases, $p_T(d,a)$ decreases.

Problem formulation and loss function

Suppose a maximum of J cohorts of n_c subjects are to be treated with doses selected from the set $\mathcal{D} = \{d_1, \ldots, d_m\}$.

At the end of the study, with data x_J, a dose $d^* \in \mathcal{D}$ will be selected as the estimate of the maximum tolerated dose.

We define the loss function

$$L(d^*, a, x_J) \;=\; |p_T(d^*, a) - 0.3| + \delta\, n_T,$$

where n_T is the number of DLEs that occur in the trial.

We place a prior $\pi(a)$ on the dose response model parameter a.

Our aim is to minimise the expected loss

$$\int (E\{|p_T(d^*, a) - 0.3|\} + \delta\, n_T)\, \pi(a)\, da.$$

over possible final decision rules and dose allocation rules that satisfy specified constraints on dose escalation.

Optimising by Dynamic Programming

The state variable

After j cohorts of n_c have been observed, the data so far comprise

$$\{U_1, Y_1, \ldots, U_j, Y_j\},$$

where $U_j \in \{1, \ldots, m\}$ is the index of the dose chosen for cohort j and Y_j the number of DLEs observed in cohort j.

It is (usually) sufficient to summarise these data by the state variable

$$X_j = (N_{1,j}, \ldots, N_{m,j}, V_{1,j}, \ldots, V_{m,j})$$

where $N_{i,j}$ is the total number of cohorts allocated dose d_i and $V_{i,j}$ the total number of DLEs observed at that dose.

With 10 cohorts of 3 and 6 doses, there are 16 million possible states at the final analysis and 25 million over the whole trial.

Optimising dose allocation in a First in Human trial

The posterior distribution of a

For data $x_j = (n_{1,j}, \ldots, n_{m,j}, v_{1,j}, \ldots, v_{m,j})$, the likelihood is

$$l(a, x_j) \;=\; \prod_{i=1}^{m} \binom{n_{i,j}}{v_{i,j}} p_i(a)^{v_{i,j}} \left\{1 - p_i(a)\right\}^{(n_{i,j} - v_{i,j})},$$

where

$$p_i(a) \;=\; \left(\frac{\tanh(d_i) + 1}{2}\right)^{a}, \quad i = 1, \ldots, m,$$

are the probabilities of a safety event at each dose d_i under model parameter a.

The posterior distribution of a has density

$$\pi_{A|X_j}(a|x_j) \;\propto\; \pi(a)\, l(a, x_j).$$

This formula does not simplify but we observe that the posterior distributions resemble Gamma distributions.

After cohort J: Estimating the Maximum Tolerated Dose

In order to minimise the overall expected loss, we must minimise the conditional expected loss

$$\int (|p_T(d^*, a) - 0.3| + \delta\, n_T)\, \pi_{A|X_J}(a|x_J)\, da.$$

If we say the MTD is d_i, the conditional expected loss is

$$\omega_{J,\,i}(x_J) \;=\; \int (|p_T(d_i, a) - 0.3| + \delta\, n_T)\, \pi_{A|X_J}(a|x_J)\, da.$$

So, we choose the dose $d_J^*(x_J)$ with the smallest value of $\omega_{J,\,i}(x_J)$.

Finally, we note that the minimum possible conditional expected loss when in state x_J is

$$\beta^{(J)}(x_J) \;=\; \min_{i \in \{1,\ldots,m\}} \omega_{J,\,i}(x_J).$$

Dynamic Programming

After cohort j: Selecting the dose for cohort $j+1$

If the current state is x_j and dose d_i is allocated to cohort j, we denote the probability of moving to state x_{j+1} under dose response model parameter a by

$$q(x_{j+1} \mid x_j, d_i, a).$$

The conditional expected loss from state x_j when dose d_i allocated to cohort j is

$$\omega_{j,i}(x_j) = \int \left\{ \sum_{x_{j+1}} q(x_{j+1} \mid x_j, d_i, a)\, \beta^{(j+1)}(x_{j+1}) \right\} \pi_{A|X_j}(a|x_j)\, da.$$

The optimal dose in state x_j is that with the smallest $\omega_{j,i}(x_j)$.

The minimum conditional expected loss when in state x_j is

$$\beta^{(j)}(x_j) = \min_{i \in \{1,\ldots,m\}} \omega_{j,i}(x_j).$$

Dynamic Programming: Implementation

Data storage

At stage j, the possible dose allocations, $(n_{1,j}, \ldots, n_{m,j})$, can be described using "stars and bars" notation (Feller, 1968), e.g.,

$$**|**||*|*|* \quad \text{for } n_{1,j} = n_{2,j} = 2,\ n_{4,j} = n_{5,j} = n_{6,j} = 1.$$

We enumerate the

$$\binom{j+m-1}{m-1}$$

vectors $(n_{1,j}, \ldots, n_{m,j})$ according to the positions of the "bars".

For each vector $(n_{1,j}, \ldots, n_{m,j})$, we list the

$$\prod_{i=1}^{m} (n_{i,j}\, n_c + 1)$$

possibilities for the vector of DLE counts $(v_{1,j}, \ldots, v_{m,j})$.

In R, we store the set of possible states as a nested list.

Dynamic Programming: Implementation

Numerical integration

Calculations involve integration over the posterior density of a for each state x_j and each possible dose d_i for cohort $j+1$.

To do this efficiently, we use Simpson's rule on a pre-prepared grid that is suitable for all posterior distributions .

Parts of the calculation that are used repeatedly, e.g., in assessing different doses for cohort $j+1$ when in state x_j, should be carried out only once.

Parallel processing

When finding the optimal dose for cohort $j+1$, calculations for different states x_j can be distributed across processors.

Example

Consider a trial with 9 cohorts of 3 subjects, and 6 dose levels.

Suppose the aim is to select the dose at which the probability of a DLE is as close as possible to 0.3.

Dose levels $d_1, \ldots, d_6$ are chosen so that, if $a = 1$, the probabilities of a DLE are 0.05, 0.1, 0.2, 0.3, 0.5 and 0.7.

The prior distribution for the dose response model parameter is

$$a \sim \text{Exp}(1).$$

We consider two loss functions which involve the dose d^* selected at the end of the trial and n_T, the number of DLEs during the trial.

Standard loss function:

$$L_{SL}(d^*, a, x_J) = |p_T(d^*, a) - 0.3|$$

Penalised loss function:

$$L_P(d^*, a, x_J) = |p_T(d^*, a) - 0.3| + 0.004\, n_T.$$

Example: Three designs

Dynamic Programming with standard loss: DP-SL

The design, found by Dynamic Programming, that minimises

$$E\{L_{SL}(d^*, a, x_J)\} \;=\; E\{|p_T(d^*, a) - 0.3|\}$$

Dynamic Programming with penalised loss: DP-P

The design, found by Dynamic Programming, that minimises

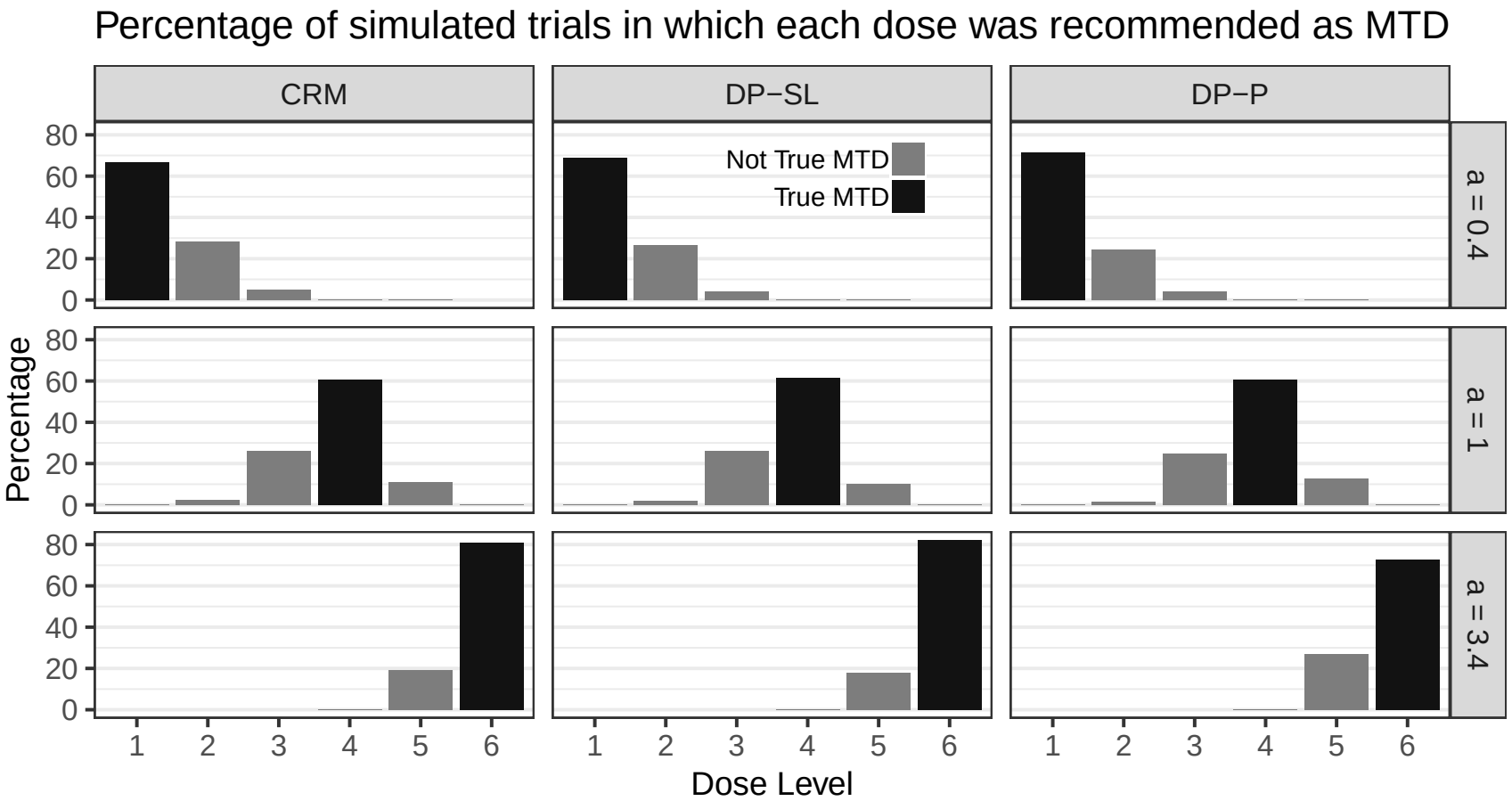
$$E\{L_P(d^*, a, x_J)\} \;=\; E\{|p_T(d^*, a) - 0.3| + 0.004\, n_T\}$$

O'Quigley et al's Continuous Reassessment Method: CRM

When in state x_j, the next cohort of patients is allocated the dose which minimises the expected value of $|p_T(d_i, a) - 0.3|$ under the posterior distribution of a,

$$\int |p_T(d_i, a) - 0.3|\, \pi_{A|X_j}(a|x_j)\, da.$$

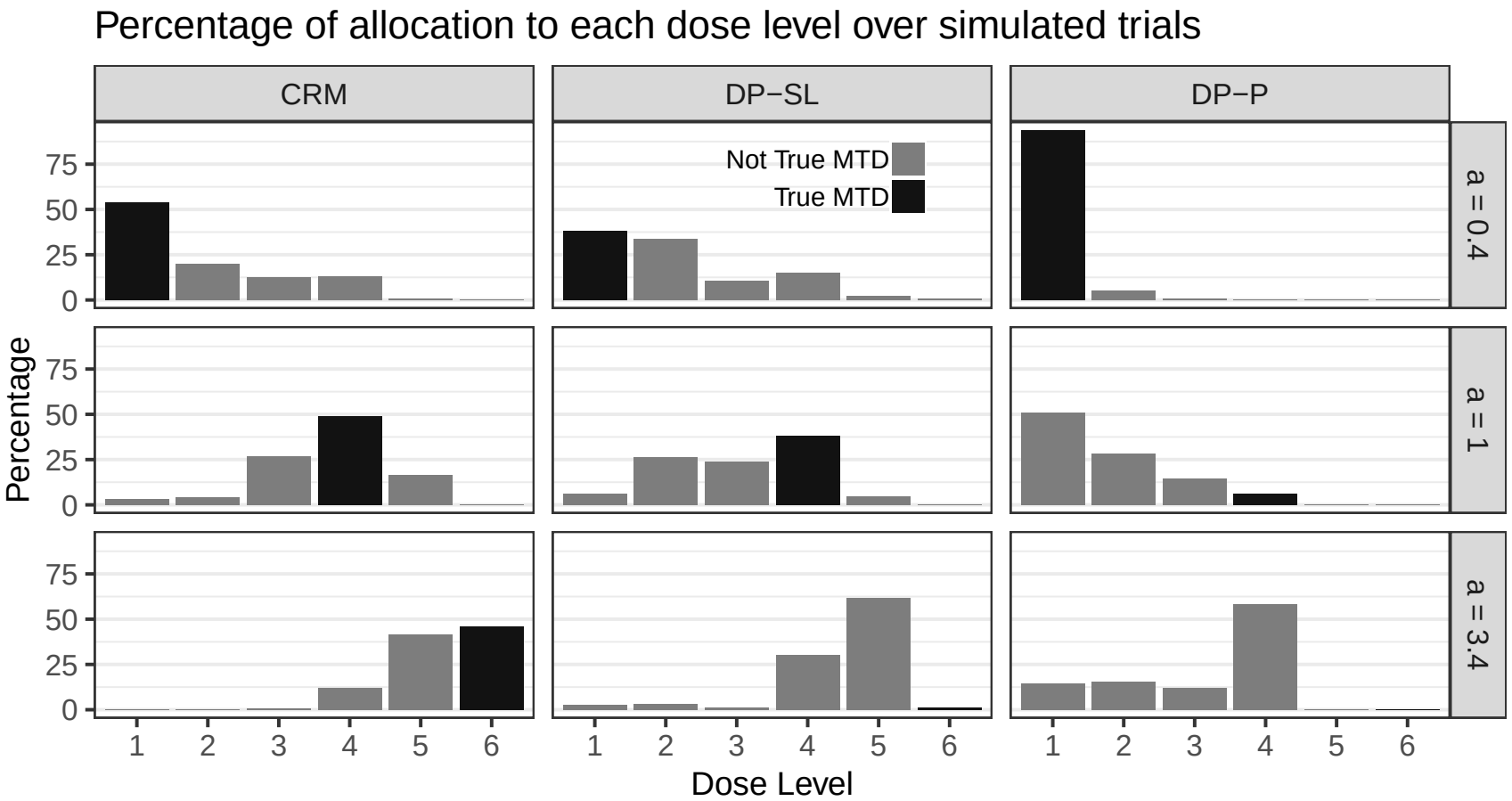
Example



$E(\text{Loss}) = 0.154$ $E(\text{Loss}) = 0.153$ $E(\text{Loss}) = 0.155$

Here "Loss" is the Standard Loss, $E\{|p_T(d^*, a) - 0.3|\}$, and expectation is over the prior distribution $a \sim \text{Exp}(1)$.

Example



$E(\text{Number of DLEs}) = 10.0$

$E(\text{Loss}) = 0.195$

$E(\text{Number of DLEs}) = 7.5$

$E(\text{Loss}) = 0.185$

Here "Loss" is the Penalised Loss, $E\{|p_T(d^*, a) - 0.3| + 0.004\, n_T\}$, and expectation is over the prior distribution $a \sim \text{Exp}(1)$.

Example: Comments

With the Standard Loss function:

O'Quigley et al's CRM is close to optimal.

But the optimised design allocates more cohorts to lower doses.

With the Penalised Loss function:

The optimised design allocates more cohorts to lower doses, reducing the number of dose limiting events.

This has little effect on the accuracy of the final decision.

Adding constraints:

It is just as easy to find the optimal design with constraints, e.g.,

The dose may increase by at most one level between cohorts,

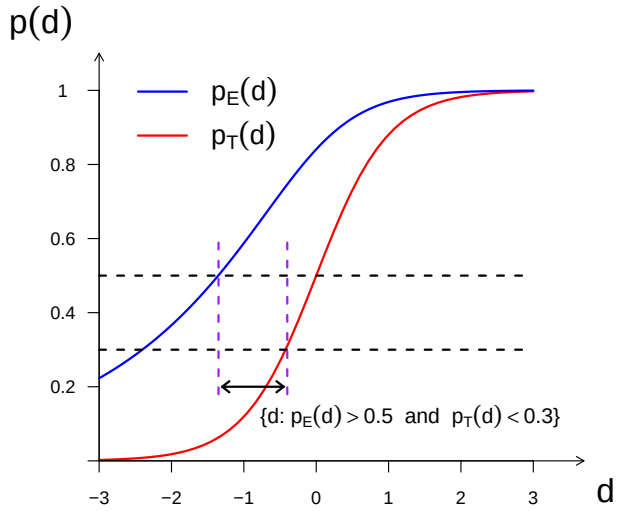
The selected dose must have been observed during the trial.

First in Human trials with safety and efficacy endpoints

Alun Bedding had posed the problem of optimising the design of a trial with safety and efficacy endpoints.

Suppose an efficacy biomarker is represented by a binary variable.

We can define a dose response model with parameter a for safety, and a model with parameter b for efficacy.



Aiming for $p_E(d) > 0.5$ and $p_T(d) < 0.3$, define the gain function

$$G(d, a, b) = I\{p_E(d) > 0.5,\, p_T(d) < 0.3\}\,\{(p_E(d) - 0.5) + (0.3 - p_T(d))\}$$

Optimising by Dynamic Programming

The state variable

After observing j cohorts, the data are summarised by

$$X_j = (N_{1,j}, \ldots, N_{m,j}, V_{1,j}, \ldots, V_{m,j}, W_{1,j}, \ldots, W_{m,j}),$$

where the additional variables $W_{i,j}$ are the numbers of positive efficacy outcomes observed at dose i.

Prior and posterior distributions of a and b

We assume independent priors

$$a \sim \text{Exp}(1) \text{ and } b \sim \text{Exp}(1)$$

and we suppose safety and efficacy events occur independently.

Optimising the design

In principle, Dynamic Programming can be applied to find the dose allocation rule that maximises the expected gain.

Approximate Dynamic Programming

Computational complexity

With two response variables, the computational demands increase dramatically.

Dynamic Programming is feasible for up to 4 cohorts of 3 — beyond that, we need a way to speed up the calculations.

The important property of any state x_j is the resulting posterior distribution of the model parameters a and b.

Thus, we shall replace the state space $\{x_j\}$ by the set of posterior distributions for (a, b).

Our assumptions imply that the posterior distributions of a and b are independent.

Making the approximation that these are Gamma distributions, a state x_j is replaced by the four parameters $(\lambda_T, k_T, \lambda_E, k_E)$ and the set of posterior distributions for (a, b) is a subset of $\mathbb{R}^4$.

Approximate Dynamic Programming

We find the optimal decision after cohort J or the optimal dose allocation after cohort $j < J$ for a given "state" $(\lambda_T, k_T, \lambda_E, k_E)$.

Working through stages $j = J,\ J-1,\ \ldots,1$, we find approximate conditional expected gain functions $\tilde{\beta}^{(j)}(\lambda_T, k_T, \lambda_E, k_E)$.

After cohort J

For a sample of states x_J:

For $i = 1, \ldots, m$, compute $\omega_{J,i}(x_J)$, the conditional expected gain when choosing d_i.

Approximate the posterior distributions of a and b given x_J as

$$a \sim \text{Gamma}(\lambda_T, k_T) \ \text{ and } \ b \sim \text{Gamma}(\lambda_E, k_E).$$

Fit Generalised Additive Models to the $\omega_{J,i}(x_J)$, giving

$$\tilde{\omega}_{J,i}(\lambda_T, k_T, \lambda_E, k_E), \quad i = 1, \ldots, m.$$

Set $\tilde{\beta}^{(J)}(\lambda_T, k_T, \lambda_E, k_E) = \max_{i \in \{1,\ldots,m\}} \tilde{\omega}_{J,i}(\lambda_T, k_T, \lambda_E, k_E)$.

Approximate Dynamic Programming

We work backwards through stages $j = J-1, \ldots, 1$.

After cohort j

For a sample of states x_j:

For $i = 1, \ldots, m$, compute $\omega_{j,i}(x_j)$, using the previously computed function $\tilde{\beta}^{(j+1)}$.

Approximate the posterior distributions of a and b given x_j as

$$a \sim \text{Gamma}(\lambda_T, k_T) \text{ and } b \sim \text{Gamma}(\lambda_E, k_E).$$

Fit Generalised Additive Models to the $\omega_{j,i}(x_j)$, giving

$$\tilde{\omega}_{j,i}(\lambda_T, k_T, \lambda_E, k_E), \quad i = 1, \ldots, m.$$

Set $\tilde{\beta}^{(j)}(\lambda_T, k_T, \lambda_E, k_E) = \max_{i \in \{1,\ldots,m\}} \tilde{\omega}_{j,i}(\lambda_T, k_T, \lambda_E, k_E)$.

First in Human trials with safety and efficacy endpoints

Lizzi has successfully implemented the approximate Dynamic Programming method for trials with 10, 12, 14 and 16 cohorts of 3.

The scale of these problems is noteworthy: with 14 cohorts of 3, the number of possible data sets is 2.8×10^{13}.

In discussion with investigators who have conducted such trials, we have found there to be significant challenges in defining trial objectives and, hence, formulating an optimisation problem.

The choice of loss or gain function can have a substantial impact on the resulting design.

So, we need to ask:

What is the goal of a Phase I trial?

How do you measure the success of such a trial?

Conclusions

Dynamic Programming is a powerful and versatile technique.

It can produce optimal designs that:

- Serve as a benchmark to assess other designs — which may have other desirable properties
- Help assess the usefulness of certain types of trial design.

Having the ability to find an optimal design given certain assumptions and objectives can reveal the need to clarify the assumptions and objectives of a proposed study.