12. Expected value of partial information

Anna Heath

Sick Kids Hospital & Dalla Lana School of Public Health | University of Toronto

anna.heath@sickkids.ca
https://sites.google.com/site/annaheathstats/
https://egon.stats.ucl.ac.uk/research/statistics-health-economics/
https://github.com/annaheath
https://github.com/StatisticsHealthEconomics

Bayesian Methods in Health Economics, Lausanne

Summary

- Formally define the EVPPI
- Estimate the EVPPI by nested MC simulation
- Introduce the regression based EVPPI estimation method
- Computing the EVPPI in R and BCEA
- Online tools for the EVPPI
  - SAVI
  - BCEAWeb

References

Bayesian Methods in Health Economics, chapter 5.4 Book website (CRC) Book website Code

Evidence Synthesis for Decision Making in Healthcare, chapter 12 Book website

Bayesian Cost-Effectiveness Analysis with the R package BCEA, chapter 4.3 Book website (Springer) Book website

Expected Value of Partial Perfect Information

- Recall that the model parameters are denoted $\boldsymbol{\theta}$ and have a distribution $p(\boldsymbol{\theta})$
- $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_P) = (\boldsymbol{\phi}, \boldsymbol{\psi})$
- What is the expected value of learning subset of parameters $\boldsymbol{\phi}$?
- Again, baseline decision – choose d with largest expected net benefit

$$\max_t \mathrm{E}_{\boldsymbol{\theta}} \left[\mathrm{NB}_t(\boldsymbol{\theta}) \right]$$

Expected Value of Partial Perfect Information

- $\boldsymbol{\theta} =$ all the model parameters; can be split into two subsets
  - The "**parameters of interest**", ϕ, e.g. prevalence of a disease, HRQL measures, length of stay in hospital, ...
  - The "**remaining parameters**, ψ, e.g. cost of treatment with other established medications, ...
- We are interested in quantifying the value of gaining more information on ϕ, while leaving current level of uncertainty on ψ unchanged

Expected Value of Partial Perfect Information

- $\boldsymbol{\theta} =$ all the model parameters; can be split into two subsets
  - The "**parameters of interest**", ϕ, e.g. prevalence of a disease, HRQL measures, length of stay in hospital, ...
  - The "**remaining parameters**, ψ, e.g. cost of treatment with other established medications, ...
- We are interested in quantifying the value of gaining more information on ϕ, while leaving current level of uncertainty on ψ unchanged

- First, consider the expected utility (EU) **if** we were able to learn ϕ but not ψ

$$\mathrm{E}_{\psi|\phi}[\mathrm{NB}_t(\boldsymbol{\theta})]$$

Expected Value of Partial Perfect Information

- $\boldsymbol{\theta} =$ all the model parameters; can be split into two subsets
  - The "**parameters of interest**", ϕ, e.g. prevalence of a disease, HRQL measures, length of stay in hospital, ...
  - The "**remaining parameters**, ψ, e.g. cost of treatment with other established medications, ...
- We are interested in quantifying the value of gaining more information on ϕ, while leaving current level of uncertainty on ψ unchanged

- First, consider the expected utility (EU) **if** we were able to learn ϕ but not ψ
- **If** we knew ϕ perfectly, best decision = the maximum of this EU

$$\max_t \mathsf{E}_{\psi|\phi}[\mathsf{NB}_t(\boldsymbol{\theta})]$$

Expected Value of Partial Perfect Information

- $\boldsymbol{\theta} =$ all the model parameters; can be split into two subsets
  - The "**parameters of interest**", ϕ, e.g. prevalence of a disease, HRQL measures, length of stay in hospital, ...
  - The "**remaining parameters**, ψ, e.g. cost of treatment with other established medications, ...
- We are interested in quantifying the value of gaining more information on ϕ, while leaving current level of uncertainty on ψ unchanged

- First, consider the expected utility (EU) **if** we were able to learn ϕ but not ψ
- **If** we knew ϕ perfectly, best decision = the maximum of this EU
- Of course, we cannot know ϕ **perfectly**, so take the expected value

$$\mathrm{E}_{\phi}\left[\max_{t} \mathrm{E}_{\psi|\phi}[\mathrm{NB}_t(\boldsymbol{\theta})]\right]$$

Expected Value of Partial Perfect Information

- $\boldsymbol{\theta} =$ all the model parameters; can be split into two subsets
  - The "**parameters of interest**", ϕ, e.g. prevalence of a disease, HRQL measures, length of stay in hospital, ...
  - The "**remaining parameters**, ψ, e.g. cost of treatment with other established medications, ...
- We are interested in quantifying the value of gaining more information on ϕ, while leaving current level of uncertainty on ψ unchanged

- First, consider the expected utility (EU) **if** we were able to learn ϕ but not ψ
- **If** we knew ϕ perfectly, best decision = the maximum of this EU
- Of course, we cannot know ϕ **perfectly**, so take the expected value
- And compare this with the **maximum expected utility overall**

$$\mathrm{E}_{\phi}\left[\max_t \mathrm{E}_{\psi|\phi}[\mathrm{NB}_t(\boldsymbol{\theta})]\right] - \max_t \mathrm{E}_{\boldsymbol{\theta}}[\mathrm{NB}_t(\boldsymbol{\theta})]$$

Expected Value of Partial Perfect Information

- $\boldsymbol{\theta}$ = all the model parameters; can be split into two subsets
  - The "**parameters of interest**", ϕ, e.g. prevalence of a disease, HRQL measures, length of stay in hospital, ...
  - The "**remaining parameters**, ψ, e.g. cost of treatment with other established medications, ...
- We are interested in quantifying the value of gaining more information on ϕ, while leaving current level of uncertainty on ψ unchanged

- First, consider the expected utility (EU) **if** we were able to learn ϕ but not ψ
- **If** we knew ϕ perfectly, best decision = the maximum of this EU
- Of course, we cannot know ϕ **perfectly**, so take the expected value
- And compare this with the **maximum expected utility overall**
- This is the EVPPI

$$\text{EVPPI} = \text{E}_{\phi}\left[\max_t \text{E}_{\psi|\phi}[\text{NB}_t(\boldsymbol{\theta})]\right] - \max_t \text{E}_{\boldsymbol{\theta}}[\text{NB}_t(\boldsymbol{\theta})]$$

Expected Value of Partial Perfect Information

- $\boldsymbol{\theta}$ = all the model parameters; can be split into two subsets
  - The "**parameters of interest**", ϕ, e.g. prevalence of a disease, HRQL measures, length of stay in hospital, ...
  - The "**remaining parameters**, ψ, e.g. cost of treatment with other established medications, ...
- We are interested in quantifying the value of gaining more information on ϕ, while leaving current level of uncertainty on ψ unchanged

- First, consider the expected utility (EU) **if** we were able to learn ϕ but not ψ
- **If** we knew ϕ perfectly, best decision = the maximum of this EU
- Of course, we cannot know ϕ **perfectly**, so take the expected value
- And compare this with the **maximum expected utility overall**
- This is the EVPPI

$$\text{EVPPI} = \text{E}_{\phi}\left[\max_t \text{E}_{\psi|\phi}[\text{NB}_t(\boldsymbol{\theta})]\right] - \max_t \text{E}_{\boldsymbol{\theta}}[\text{NB}_t(\boldsymbol{\theta})]$$

- **That** is the difficult part!
  - Can do nested Monte Carlo, but takes for ever to get accurate results
  - **Recent methods** based on **GAMs/Gaussian Process regression/spatial modelling** very efficient and quick!

doi Strong et al (2014) and doi Heath et al (2016)

Expected Value of Partial Perfect Information

$$\text{EVPPI} = \text{E}_{\phi}\left[\max_t \text{E}_{\psi|\phi}[\text{NB}_t(\boldsymbol{\theta})]\right] - \max_t \text{E}_{\boldsymbol{\theta}}[\text{NB}_t(\boldsymbol{\theta})]$$

- Can rewrite as

$$\begin{aligned}\text{EVPPI}(\boldsymbol{\phi}) &= \text{E}_{\phi}\left[\max_t \text{E}_{\psi|\phi}\left\{\text{NB}_t(\boldsymbol{\phi},\boldsymbol{\psi})\right\}\right] - \max_t \text{E}_{\boldsymbol{\theta}}\left[\text{NB}_t(\boldsymbol{\theta})\right] \\ &= \text{E}_{\phi}\left[\max_t \text{E}_{\psi|\phi}\left\{\text{NB}_t(\boldsymbol{\phi},\boldsymbol{\psi})\right\}\right] - \max_t \text{E}_{\phi}\left[\text{E}_{\psi|\phi}\left\{\text{NB}_t(\boldsymbol{\theta})\right\}\right]\end{aligned}$$

- Here, the second term is estimated using the same MC simulation as the first

Problems calculating the EVPPI by MC simulation

EVPPI for the parameters of interest ϕ

$$\text{EVPPI}(\boldsymbol{\phi}) = \text{E}_{\boldsymbol{\phi}}\left[\max_t \text{E}_{\boldsymbol{\psi}|\boldsymbol{\phi}}\left\{\text{NB}_t(\boldsymbol{\phi}, \boldsymbol{\psi})\right\}\right] - \max_t \text{E}_{\boldsymbol{\phi}}\left[\text{E}_{\boldsymbol{\psi}|\boldsymbol{\phi}}\left\{\text{NB}_t(\boldsymbol{\theta})\right\}\right]$$

where the parameters $\boldsymbol{\theta}$ are partitioned into those of interest $\boldsymbol{\phi}$ and the rest $\boldsymbol{\psi}$

- These inner expectations are problematic
  - Nested expectations are potentially slow to evaluate by Monte Carlo
  - $p(\boldsymbol{\psi} \mid \boldsymbol{\phi})$ could be a difficult conditional distribution to sample from
  - May require MCMC (using e.g. BUGS, JAGS, bespoke code, ...)
- If $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are independent then $p(\boldsymbol{\psi} \mid \boldsymbol{\phi}) = p(\boldsymbol{\psi})$, which makes things easier

Estimating the EVPPI

1. Use analytic expressions, or approximations, to the inner expectation
2. Methods that only work for single parameters:
    - Strong and Oakley (2013)
    - Sadatsafavi et al (2013)

Estimating the EVPPI

1. Use analytic expressions, or approximations, to the inner expectation
2. Methods that only work for single parameters:
   - Strong and Oakley (2013)
   - Sadatsafavi et al (2013)
3. Regression-based methods that work for any number of parameters

EVPPI for the parameters of interest ϕ

$$\text{EVPPI}(\boldsymbol{\phi}) = \text{E}_{\boldsymbol{\phi}}\left[\max_t \text{E}_{\boldsymbol{\psi}|\boldsymbol{\phi}}\left\{\text{NB}_t(\boldsymbol{\phi},\boldsymbol{\psi})\right\}\right] - \max_t \text{E}_{\boldsymbol{\phi}}\left[\text{E}_{\boldsymbol{\psi}|\boldsymbol{\phi}}\left\{\text{NB}_t(\boldsymbol{\theta})\right\}\right]$$

Regression equation

$$\begin{aligned}\text{NB}_t(\boldsymbol{\theta}) &= \text{E}_{\boldsymbol{\psi}|\boldsymbol{\phi}}\left[\text{NB}_t(\boldsymbol{\phi},\boldsymbol{\psi})\right] + \varepsilon \\ &= g_t(\boldsymbol{\phi}) + \varepsilon\end{aligned}$$

Estimating the EVPPI

1. Use analytic expressions, or approximations, to the inner expectation
2. Methods that only work for single parameters:
   - Strong and Oakley (2013)
   - Sadatsafavi et al (2013)
3. Regression-based methods that work for any number of parameters

EVPPI for the parameters of interest ϕ

$$\text{EVPPI}(\boldsymbol{\phi}) = \text{E}_{\boldsymbol{\phi}}\left[\max_t \text{E}_{\boldsymbol{\psi}|\boldsymbol{\phi}}\left\{\text{NB}_t(\boldsymbol{\phi},\boldsymbol{\psi})\right\}\right] - \max_t \text{E}_{\boldsymbol{\phi}}\left[\text{E}_{\boldsymbol{\psi}|\boldsymbol{\phi}}\left\{\text{NB}_t(\boldsymbol{\theta})\right\}\right]$$

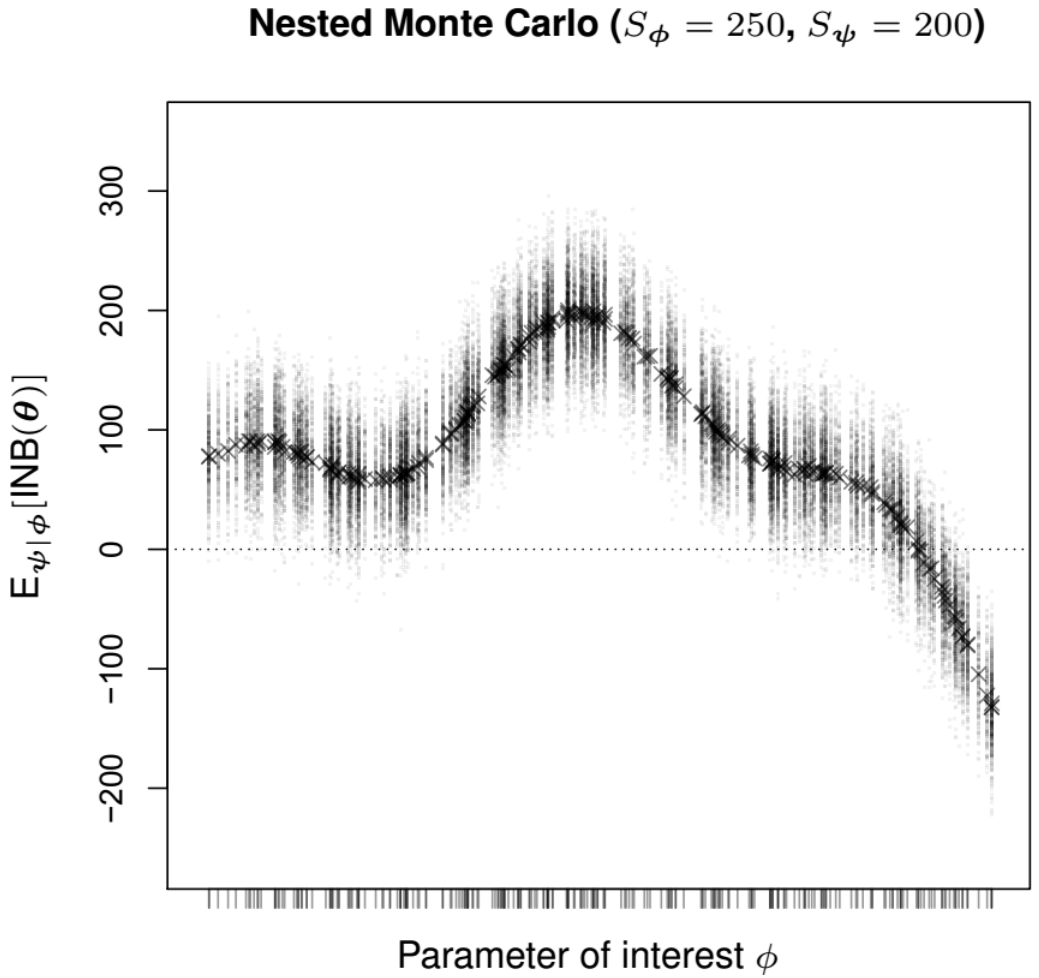
Regression equation

$$\begin{aligned}\text{NB}_t(\boldsymbol{\theta}) &= \text{E}_{\boldsymbol{\psi}|\boldsymbol{\phi}}\left[\text{NB}_t(\boldsymbol{\phi},\boldsymbol{\psi})\right] + \varepsilon \\ &= g_t(\boldsymbol{\phi}) + \varepsilon\end{aligned}$$

- $g_t(\boldsymbol{\phi})$ is some unknown *smooth* function of $\boldsymbol{\phi}$ and can show that $\text{E}\left[\varepsilon\right] = 0$
- Treat as non-parametric regression of $\text{NB}_t(\boldsymbol{\theta})$ on $\boldsymbol{\phi}$ (Strong et al 2014)

EVPPI – Brute force/nested MC

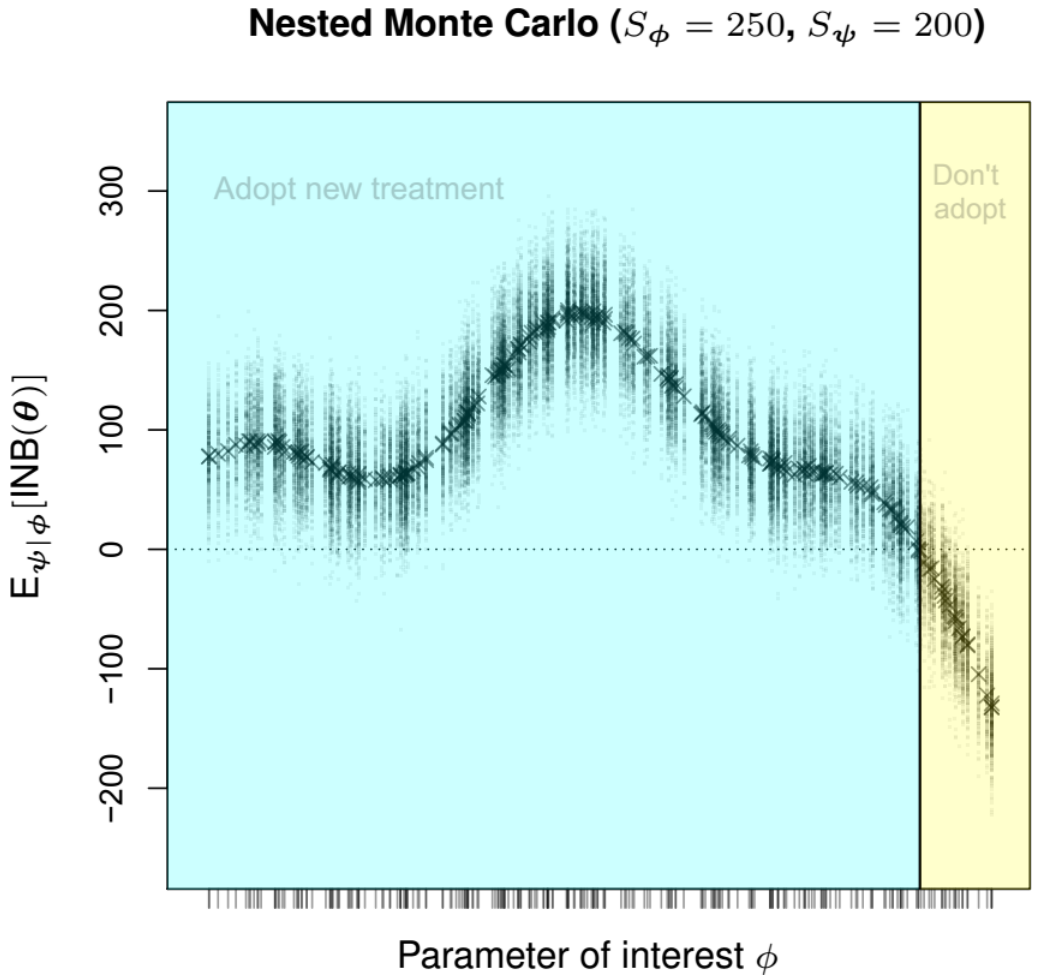
Assuming there are only two interventions, can consider $\text{IB}(\boldsymbol{\theta}) = \text{NB}_1(\boldsymbol{\theta}) - \text{NB}_0(\boldsymbol{\theta})$



Slide stolen from Mark Strong

EVPPI – Brute force/nested MC

Assuming there are only two interventions, can consider $\text{IB}(\boldsymbol{\theta}) = \text{NB}_1(\boldsymbol{\theta}) - \text{NB}_0(\boldsymbol{\theta})$



EVPPI – model as a regression problem...

Can model as a **regression** problem

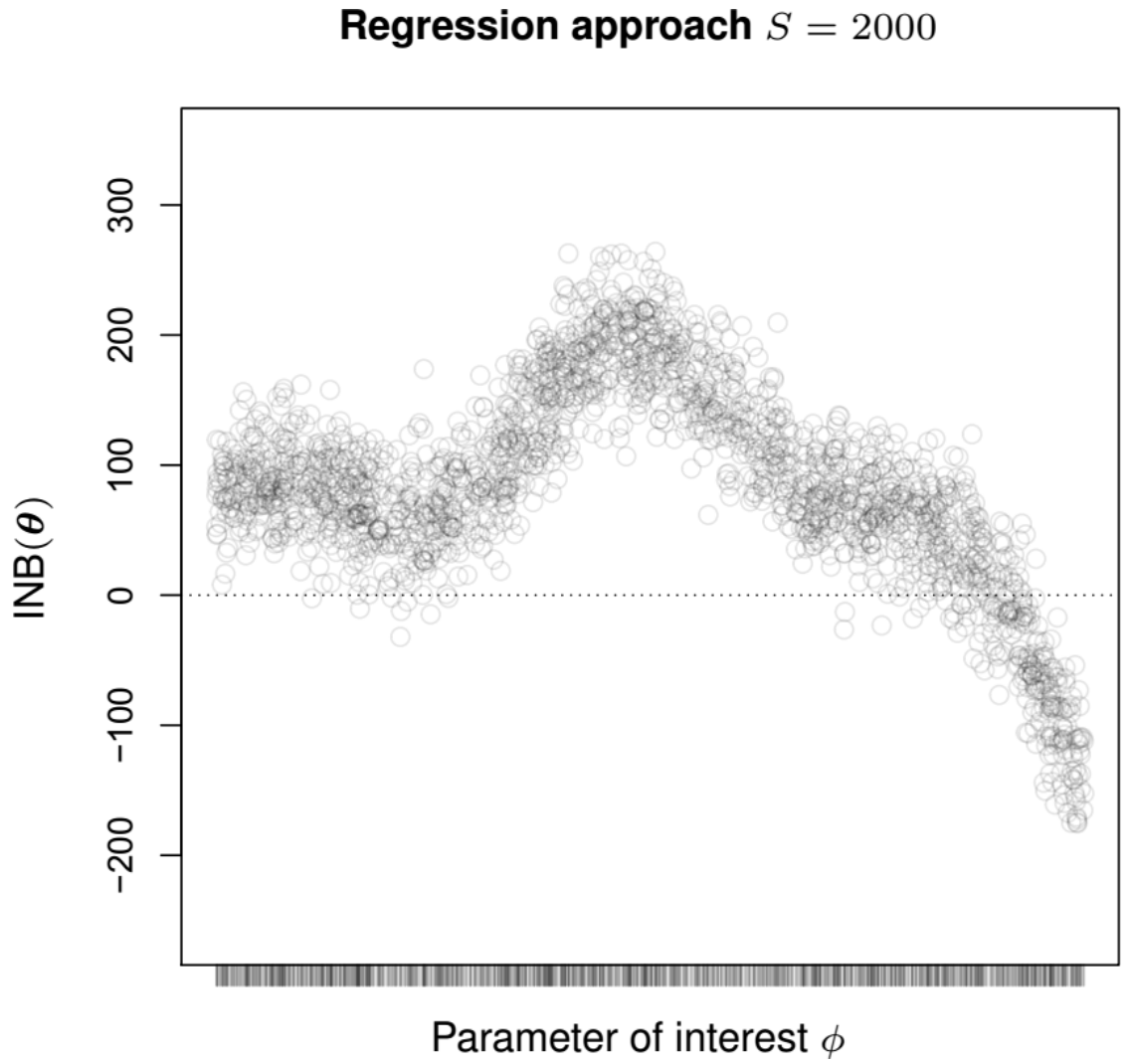
$$\begin{aligned} \text{NB}_t(\boldsymbol{\theta}) &= \text{E}_{\boldsymbol{\psi}|\boldsymbol{\theta}}[\text{NB}_t(\boldsymbol{\theta})] + \varepsilon, \qquad \text{with } \varepsilon \sim \text{Normal}(0, \sigma_\varepsilon^2) \\ &= g(\boldsymbol{\phi}) + \varepsilon \end{aligned}$$

"Data"

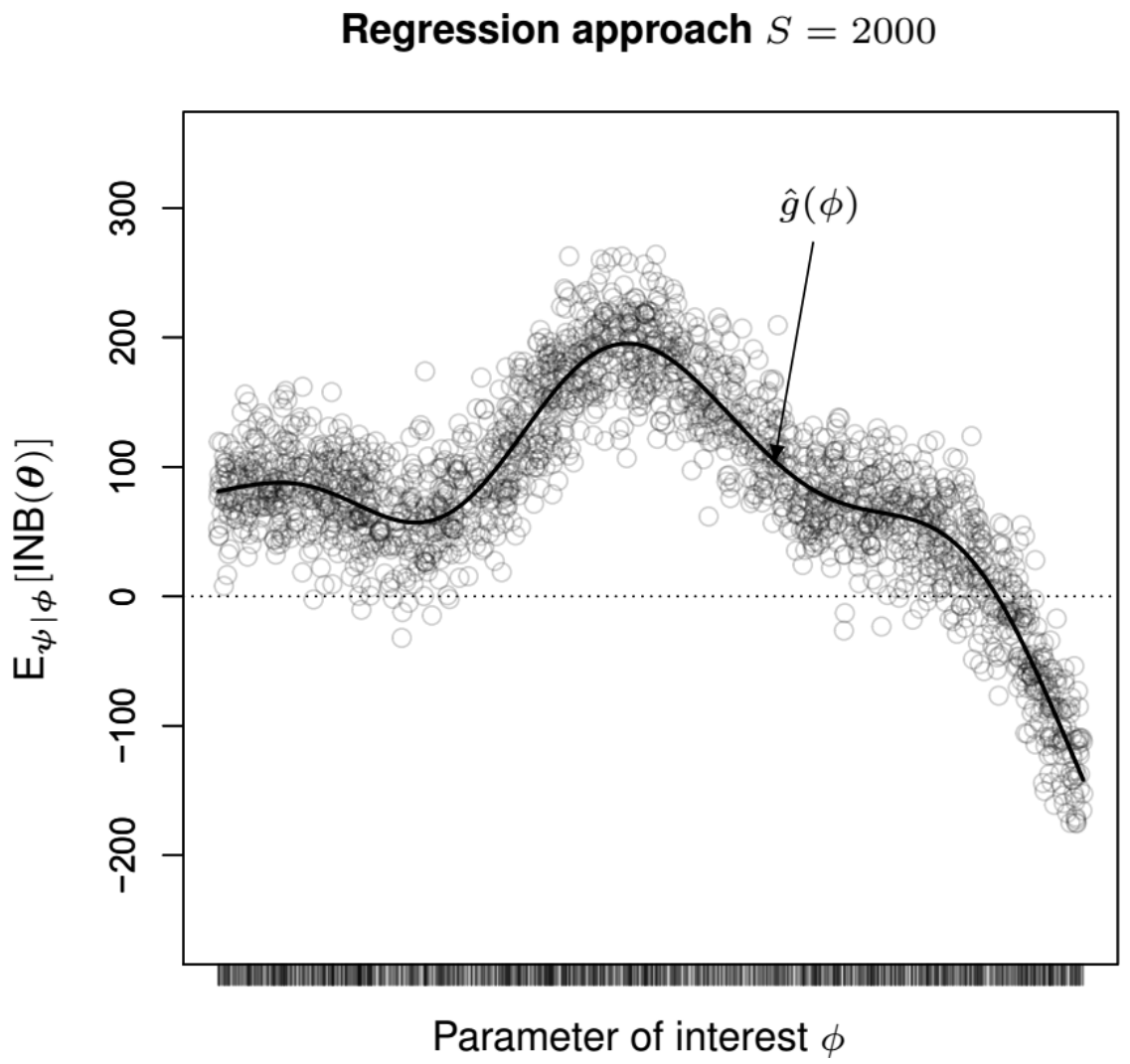
- **Simulations** for $\text{NB}_t(\boldsymbol{\theta})$ as "response"
- **Simulations** for $\boldsymbol{\phi}$ as "covariates"
- **NB**: Only need S data points (=PSA simulations), instead of $S_{\boldsymbol{\phi}} \times S_{\boldsymbol{\psi}}$!

	Parameter simulations ('covariates')				'Responses'	
Iteration	π_0	ρ	...	γ	$\text{NB}_0(\boldsymbol{\theta})$	$\text{NB}_1(\boldsymbol{\theta})$
1	0.585	0.3814	...	0.4194	77480	67795
2	0.515	0.0166	...	0.0768	87165	106535
3	0.611	0.1373	...	0.0592	58110	38740
4	0.195	0.7282	...	0.7314	77480	87165
...	...	...	...	...	...	...
S	0.0305	0.204	...	0.558	48425	87165

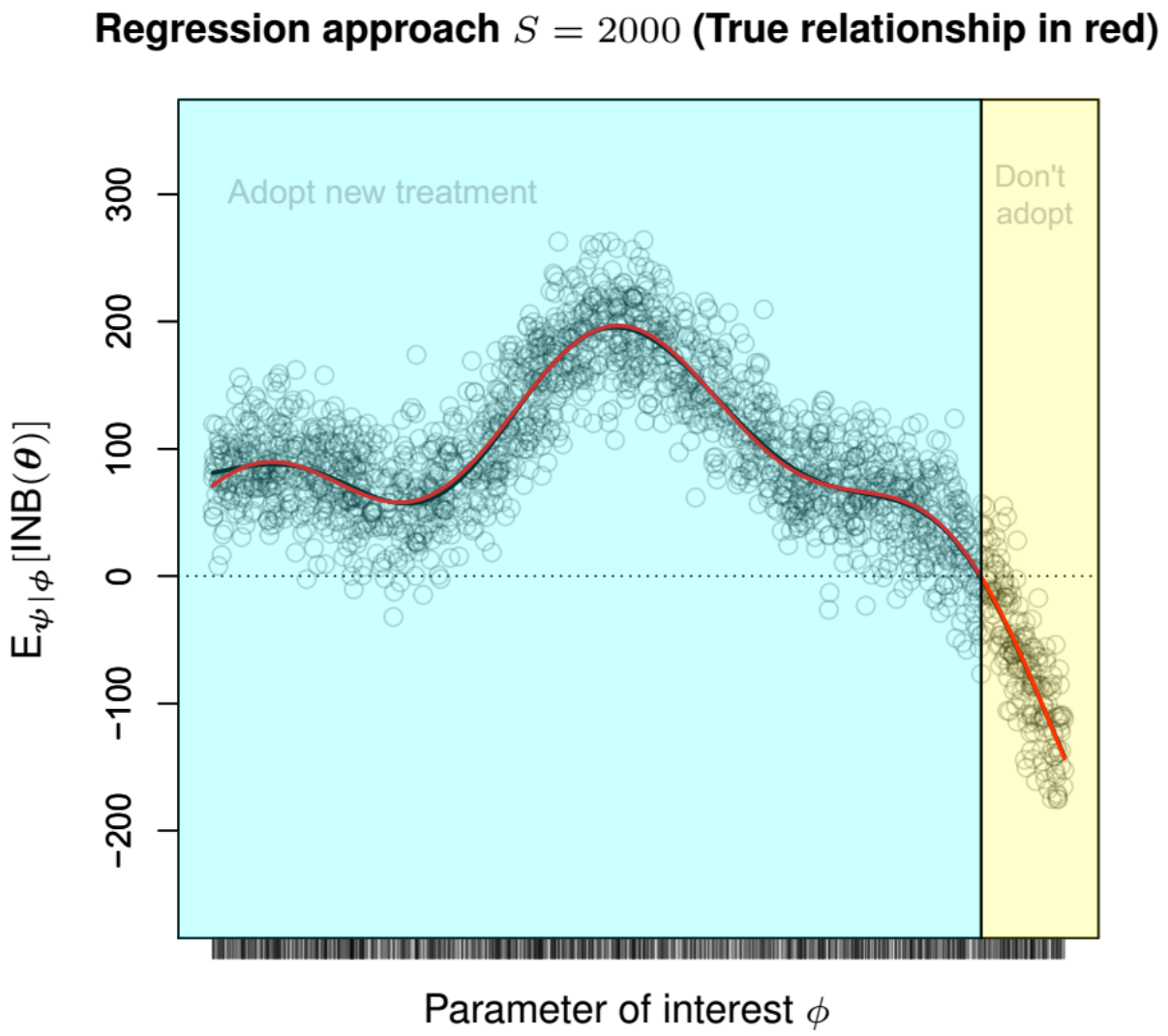
EVPPI – model as a regression problem...



EVPPI – model as a regression problem...



EVPPI – model as a regression problem...



EVPPI – model as a regression problem...

- Once the functions $g_t(\boldsymbol{\phi})$ are estimated, can approximate

$$\begin{aligned}\text{EVPPI} &= \mathsf{E}_{\boldsymbol{\phi}}\left[\max_t \mathsf{E}_{\boldsymbol{\psi}|\boldsymbol{\phi}}[\text{NB}_t(\boldsymbol{\theta})]\right] - \max_t \mathsf{E}_{\boldsymbol{\theta}}[\text{NB}_t(\boldsymbol{\theta})] \\ &\approx \frac{1}{S}\sum_{s=1}^{S} \max_t \hat{g}_t(\boldsymbol{\phi}_s) - \max_t \frac{1}{S}\sum_{s=1}^{S} \hat{g}_t(\boldsymbol{\phi}_s)\end{aligned}$$

EVPPI – model as a regression problem...

- Once the functions $g_t(\boldsymbol{\phi})$ are estimated, can approximate

$$\begin{aligned}\text{EVPPI} &= \mathsf{E}_{\boldsymbol{\phi}}\left[\max_t \mathsf{E}_{\boldsymbol{\psi}|\boldsymbol{\phi}}[\text{NB}_t(\boldsymbol{\theta})]\right] - \max_t \mathsf{E}_{\boldsymbol{\theta}}[\text{NB}_t(\boldsymbol{\theta})] \\ &\approx \frac{1}{S}\sum_{s=1}^{S}\max_t \hat{g}_t(\boldsymbol{\phi}_s) - \max_t \frac{1}{S}\sum_{s=1}^{S}\hat{g}_t(\boldsymbol{\phi}_s)\end{aligned}$$

- **NB:** $g_t(\boldsymbol{\phi})$ can be complex, so need to use **flexible** regression methods
  - **GAMs:** $g_t(\boldsymbol{\phi}) = \sum_{q=1}^{Q_\phi} h_t(\phi_{sq})$ with $h_t(\cdot) =$ smooth functions (cubic polynomials)
  - very fast, but only if number of important parameters $Q_\phi \leq 5$ (interactions increase model size exponentially)
  - If $Q_\phi > 5$ then use **Gaussian Process** regression (GPR)
    - Strong et al: original GPR method
    - Heath et al: based on spatial modelling; can be more computationally efficient
- Other methods based on alternative approaches
  - Most are implemented in the R package BCEA (see also: https://github.com/giabaio/BCEA)

Computing EVPI via GAM regression in R directly

For example, if we wish to estimate the EVPPI of $\phi = \theta_1$

```
> library(mgcv)
> model1 <- gam(nb1 ~ s(theta1))
> model2 <- gam(nb2 ~ s(theta1))
> g.hat1 <- fitted(model1)
> g.hat2 <- fitted(model2)
> evppi <- mean(pmax(g.hat1, g.hat2)) - max(mean(g.hat1), mean(g.hat2))
```

Computing EVPI via GAM regression in R directly

For example, if we wish to estimate the EVPPI of $\phi = \theta_1$

```
> library(mgcv)
> model1 <- gam(nb1 ~ s(theta1))
> model2 <- gam(nb2 ~ s(theta1))
> g.hat1 <- fitted(model1)
> g.hat2 <- fitted(model2)
> evppi <- mean(pmax(g.hat1, g.hat2)) - max(mean(g.hat1), mean(g.hat2))
```

Or the EVPPI for a group $\phi = \{\theta_1, \theta_2, \theta_3\}$

```
> model1 <- gam(nb1 ~ te(theta1, theta2, theta3))
> model2 <- gam(nb2 ~ te(theta1, theta2, theta3))
> ...
```

- The syntax `s()` in the `gam` function specifies a thin plate regression spline
- The syntax `te()` in the `gam` function specifies a cubic spline
- Very simple R code, and fast (for further details see Strong et al 2014)

EVPPI via GAM for groups of inputs

⚠ We can sometimes run into problems in the case where we wish to estimate EVPPI for a group of inputs and have specified a multivariate smooth term, i.e.

$$\text{NB}_t(\boldsymbol{\theta}) = s(\theta_1, \ldots, \theta_q) + \varepsilon$$

- If the number of independent variables is even moderately sized (e.g. more than 5 or 6 or so), we can run into computational problems
  - The number of basis functions required to define the multivariate smooth function is N^q, where N is the number of basis functions in each input dimension
  - This means that it can be difficult to calculate EVPI for groups of inputs where the number of inputs in the group is larger than 5 or 6 or so

EVPPI via GAM for groups of inputs

- If we want to estimate the EVPPI for a large group of inputs, we could specify a model in which the smooth terms for each input were additive

$$\text{NB}_t(\boldsymbol{\theta}) = s(\theta_1) + s(\theta_2) + \ldots + \varepsilon$$

- Or, in which smooth functions of sub-groups of terms were additive

$$\text{NB}_t(\boldsymbol{\theta}) = s(\theta_1, \theta_2, \theta_3) + s(\theta_4, \theta_5, \theta + 6) + \ldots + \varepsilon$$

- How well this approximates the 'true' EVPI will depend on the nature of the interaction between variables
- Alternative non-parametric regression approaches include Gaussian Process regression, Multivariate Adaptive Regression Splines (MARS), Neural Networks, Projection Pursuit Regression, Regression Trees
- These all allow for large numbers of interactions

Non-parametric regression approach

- From the T net benefits for T decision options, we can calculate $T-1$ incremental net benefits against some chosen reference option
- Working with the incremental net benefits allows us to reduce the number of regressions from T to $T-1$

Non-parametric regression approach

- From the T net benefits for T decision options, we can calculate $T-1$ incremental net benefits against some chosen reference option
- Working with the incremental net benefits allows us to reduce the number of regressions from T to $T-1$

- For example, if we wish to estimate the EVPPI of $\phi = \theta_1$:

```
> ## "Old" code
> model1 <- gam(nb1 ~ s(theta1))
> model2 <- gam(nb2 ~ s(theta1))
> g.hat1 <- fitted(model1)
> g.hat2 <- fitted(model2)
> evppi <- mean(pmax(g.hat1, g.hat2)) - max(mean(g.hat1), mean(g.hat2))
>
> # "New" code
> model <- gam(inb ~ s(theta1))
> g.hat <- fitted(model)
> evppi <- mean(pmax(0, g.hat)) - max(0, mean(g.hat))
```

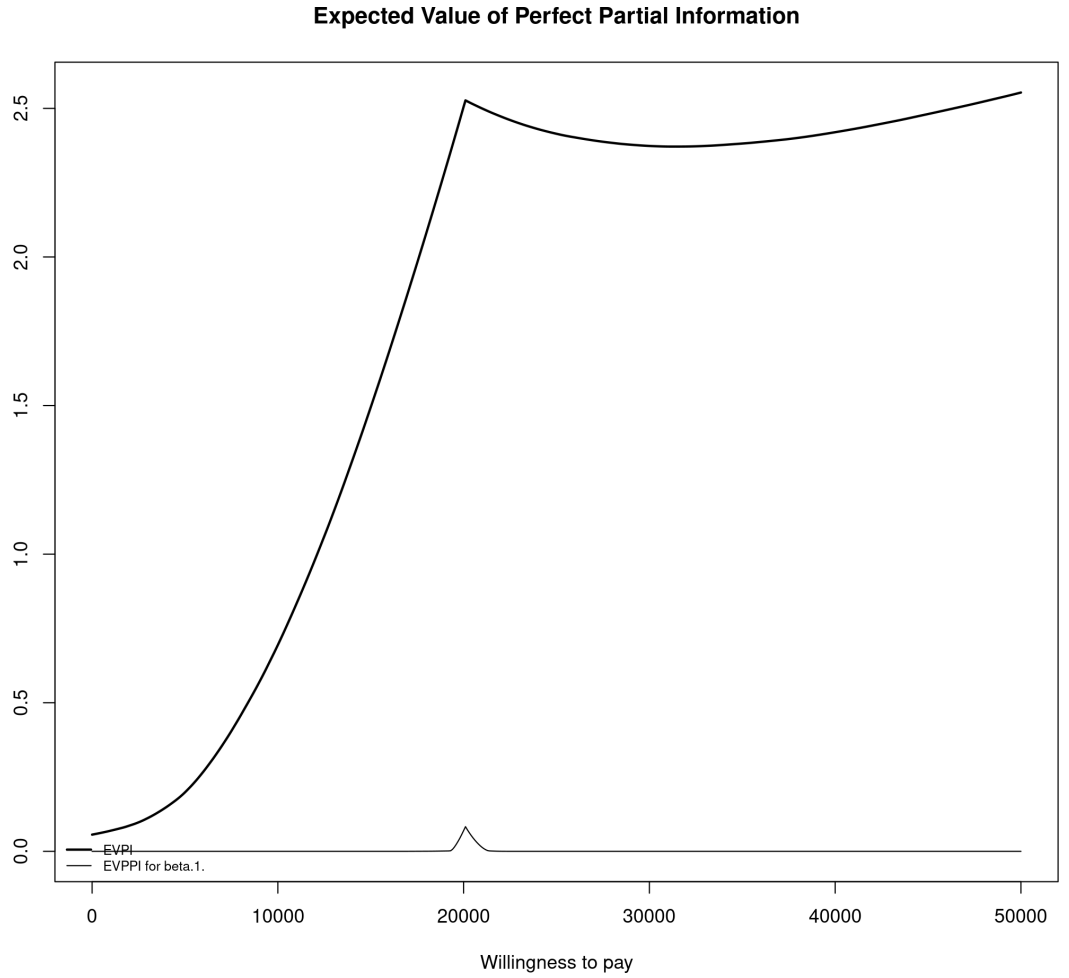
Calculating the EVPPI in BCEA

Code | Output

```
> library(BCEA)
>
> # Load the data
> data(Vaccine)
>
> # Runs BCEA to post-process the cost-effectiveness model
> m = bcea(e,c,interventions=treats,ref=2)
>
> # Formats the input
> PSA = createInputs(vaccine_mat,print_is_linear_comb = FALSE)
>
> # Computes the EVPPI using BCEA
> VoI = evppi(m,c("beta.1."),PSA$mat)
>
> # Plots the results
> plot(VoI)
```

Calculating the EVPPI in BCEA

Code | Output



Across Willingness-to-Pay

- Notice that BCEA calculates the EVPPI across willingness-to-pay
- In previous lecture, we fit the following regression curve:

$$\text{NB}_t^{(j)} = g_t\left(\boldsymbol{\phi}^{(j)}\right) + \varepsilon^{(j)}$$

where

$$\text{NB}_t^{(j)} = ke_t^{(j)} - c_t^{(j)}$$

- To calculate across k, we fit two regression curves – one for effects and one for costs
- This is important for later...

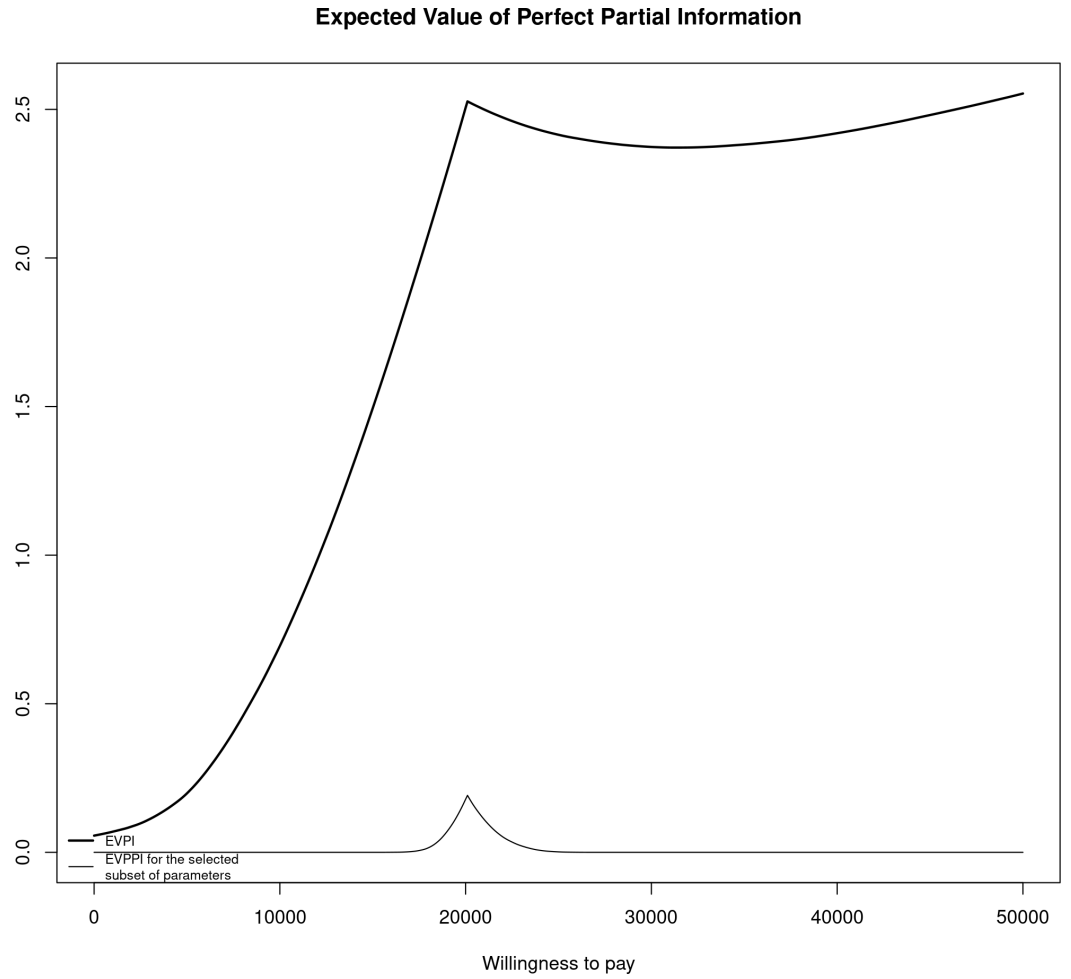
Calculating EVPPI with larger parameter subsets in BCEA

Code | Output

```
> VoI = evppi(m, c("beta.1.", "beta.2.", "beta.3.", "beta.4.", "beta.5."),PSA$mat)
> plot(VoI)
```

Calculating EVPPI with larger parameter subsets in BCEA

Code | Output



Default methods

- When calculating the EVPPI with BCEA the regression method is displayed
- For EVPPI with less than 5 parameters

  `Calculating fitted values for the GAM regression`

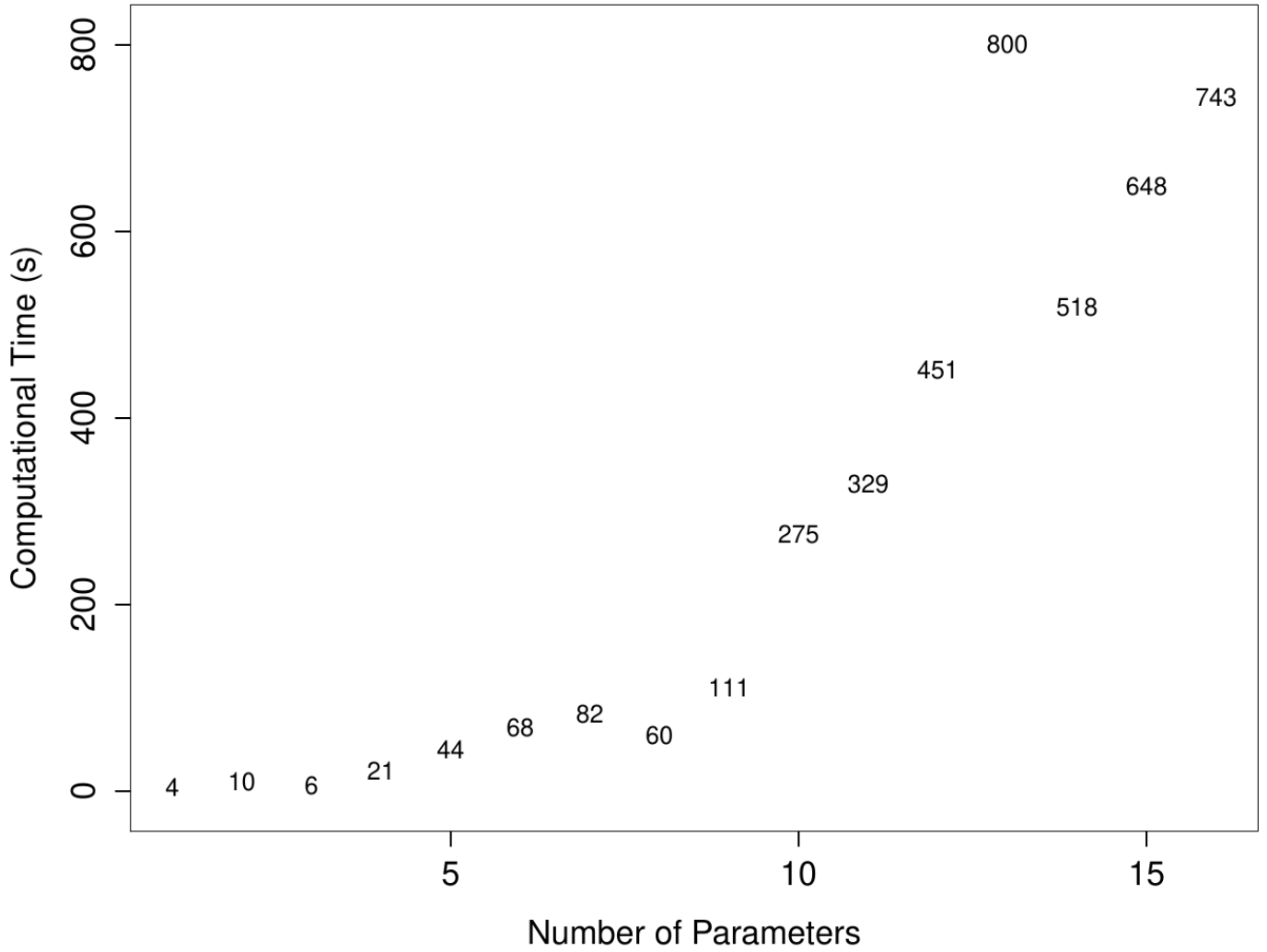
  Fast and accurate for small numbers of parameters

- For EVPPI with 5 or more parameters

  `Calculating fitted values for the GP regression using INLA/SPDE`

  Fast approximation to GP regression – recommended multi-parameter method

Why INLA/SPDE?



How does the INLA/SPDE method work?

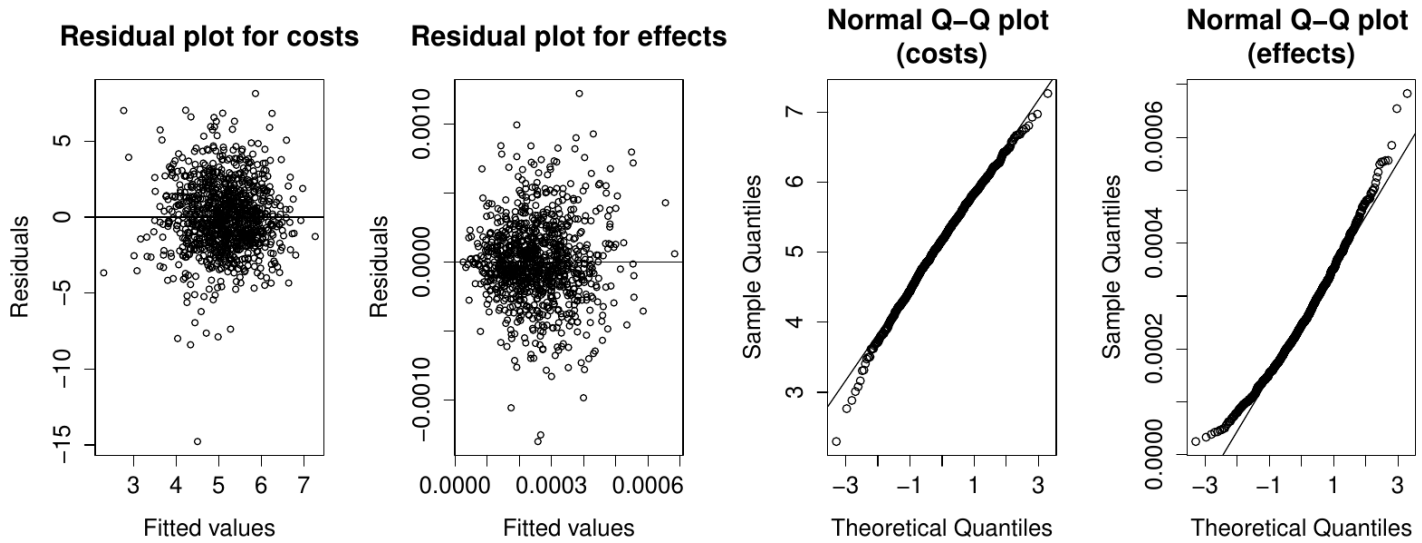
1. Starts with a linear model for all the elements in ϕ
2. *Corrects* for non-linearity in the relationship between ϕ and NB(θ) using a "mesh" `Determining Mesh`
3. Reduces the dimension of ϕ in this non-linear part `Finding projections`
4. Uses a fast Bayesian computation method (INLA)

More details in Heath et al 2016

Residual checking

- This approximation is usually sufficient to estimate the EVPPI
- We can check the accuracy of the fit by looking at residuals
- This can be performed easily in BCEA

```
> diag.evppi(VoI,m)
> diag.evppi(VoI,m,int=1,diag="qqplot")
```



Multi-decision example

- Sometime the dimension reduction is not sufficient to understand the non-linear part
- Residual checking is vital in these situations
- BCEA tells you which regression fit is potentially poorly fitted

```
> VoI<-evppi(c(1,2,3),inp$mat,m)

Warning message:
In make.proj(parameter = parameter, inputs = inputs, x = x, k = k, :
  The dimension of the sufficient reduction for the incremental effects , column 1 , is 3 .
                  Dimensions greater than 2 imply that the EVPPI approximation using INLA may be inaccurate.
  Full residual checking using diag.evppi is required.

> diag.evppi(VoI,m,int=1)
```

How to improve the approximation?

Issues with residuals

If there are issues with the residual plots, there are two main ways to improve the EVPPI estimate:

1. Include interactions in the linear part of the regression curve for the fast approximation
2. Use full GP regression – slower but more accurate

How to improve the approximation?

Issues with residuals

If there are issues with the residual plots, there are two main ways to improve the EVPPI estimate:

1. Include interactions in the linear part of the regression curve for the fast approximation
2. Use full GP regression – slower but more accurate

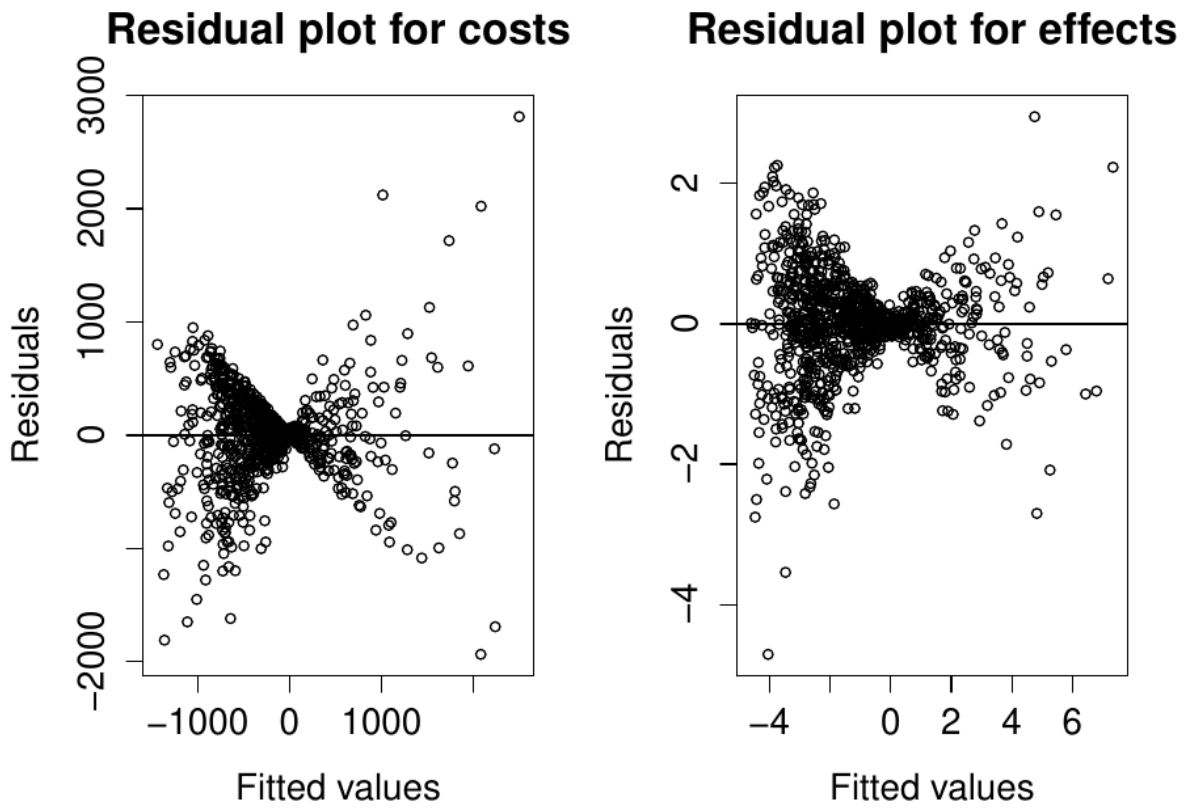
Extending the linear part

- Extending the linear structure is fast but you will still see a warning
- Residuals checks must be used to see whether the fit is accurate
- We specify the level of interactions using a list

```
> VoI<-evppi(c(1,2,3,4,5),PSA$mat,m,
   int.ord=list(c(2,1,1), #effects,
   c(2,1,1) #costs
))

> diag.evppi(VoI,m,int=1)
```

Mixed strategy



Residuals are **very** structured, so must use full GP

```
> VoI<-evppi(c(1,2,3),inp$mat,m,
   method=list(c("GP","INLA","INLA") #effects,
   c("GP","INLA","INLA")) #costs
   n.sim=1000)
```

Online tools