Tegaderm CHG IV Securement Dressing for Central Venous and Arterial Catheter Insertion Sites

A decision tree example with probabilistic sensitivity analysis

Andrew J. Sims

2020-05-07

Introduction

This vignette is an example of modelling a decision tree using the `rdecision` package, with probabilistic sensitivity analysis. It is based on the model reported by Jenks *et al* (2016) in which a transparent dressing used to secure vascular catheters (Tegaderm CHG) was compared with a standard dressing.

Model variables

The following code creates the 13 variables used in the model. The choice of variables, their distributions and their parameters are taken from table 3 of Jenks *et al* (2016), with the following corrections:

- For variables with lognormal uncertainty, the manufacturer gave values for the mean m and standard deviation s in log space. However, their standard deviations were quoted as negative values. This was an error, but had no effect on their results, because they sampled values of $\exp(m + sz)$, where z is a sample from a standard normal distribution and is symmetrical about 0. For the variables with log normal uncertainty given below, positive standard deviation parameters, with the same absolute value, have been used as hyperparameters of the log normal distributions. Also note that the *median* value on the natural scale of a random variable distributed as $logN(m, s)$, where μ and σ are the mean and standard deviation on the log scale, is e^{μ}; the mean on the natural scale is slightly larger. For example, the hazard ratio for CRBSI with Tegaderm versus standard dressing was modelled as $logN(-0.911, 0.393)$, which has median 0.402 (the point estimate of the ratio from the literature) and mean 0.434.
- The relative risk for dermatitis was modelled as $logN(1.482, 0.490)$.
- The point estimate cost of CRBSI was £9900, not £9990, although the parameters (198,50) are quoted correctly.

```
# clinical variables
r.CRBSI <- NormalModelVariable$new(
  'Baseline CRBSI rate',
  '/1000 catheter days',
  mu=1.48, sigma=0.074
)
hr.CRBSI <- LogNormalModelVariable$new(
  'Tegaderm CRBSI HR',
  'ratio',
  p1=-0.911, p2=0.393
)
r.LSI <- NormalModelVariable$new(
  'Baseline LSI rate',
  '/patient',
  mu=0.1, sigma=0.01
)
```

```
hr.LSI <- LogNormalModelVariable$new(
  'Tegaderm LSI HR',
  'ratio',
  p1=-0.911, p2=0.393
)
r.Dermatitis <- NormalModelVariable$new(
  'Baseline dermatitis risk',
  '/catheter',
  mu=0.0026, sigma=0.00026
)
rr.Dermatitis <- LogNormalModelVariable$new(
  'Tegaderm Dermatitis RR',
  'ratio',
  p1=1.482, p2=0.490
)

# cost variables
c.CRBSI <- GammaModelVariable$new('CRBSI cost', 'GBP', alpha=198.0, beta=50)
c.Dermatitis <- GammaModelVariable$new('Dermatitis cost', 'GBP', alpha=30, beta=5)
c.LSI <- GammaModelVariable$new('LSI cost', 'GBP', alpha=50, beta=5)
c.Tegaderm <- ConstModelVariable$new('Tegaderm CHG cost', 'GBP', const=6.21)
c.Standard <- ConstModelVariable$new('Standard dressing cost', 'GBP', const=1.34)
n.cathdays <- NormalModelVariable$new('No. days with catheter', 'days', mu=10, sigma=2)
n.dressings <- NormalModelVariable$new('No. dressings', 'dressings', mu=3, sigma=0.3)
```

The decision tree approach

Using `rdecision` the decision problem may be solved by constructing a decision tree comprising decision nodes, chance nodes and leaf nodes. The general approach is to create model variable expressions, which link model variables, construct a tree, and then evaluate the decision for the base case and proabilistically.

Model variable expressions

Variables in the model may be included in the decision tree via model variable expressions, which are mathematical expressions which involve model variables. The most simple form of model variable expression involving a model variable `X` is `quote(X)`. More complex forms of expression involving R's numerical functions and multiple model variables are supported, provided the expressions conform to R syntax.

The following code creates the model variable expressions to be used as values in the decision tree nodes.

```
# probabilities
p.Dermatitis.S <- ExpressionModelVariable$new(
  'Probability of dermatitis with standard dressing', 'P',
  quote(n.dressings*r.Dermatitis)
)
p.Dermatitis.T <- ExpressionModelVariable$new(
  'Probability of dermatitis with Tegaderm', 'P',
  quote(n.dressings*r.Dermatitis*rr.Dermatitis)
)
p.LSI.S <- ExpressionModelVariable$new(
 'Probability of LSI with standard dressing', 'P',
 quote(r.LSI)
```

```
)
#> Warning in .subset2(public_bind_env, "initialize")(...):
#> ExpressionModelVariable$new: expr must be of type 'call'
p.LSI.T <- ExpressionModelVariable$new(
 'Probability of LSI with Tegaderm', 'P',
 quote(r.LSI*hr.LSI)
)
p.CRBSI.S <- ExpressionModelVariable$new(
  'Probability of CRBSI with standard dressing', 'P',
  quote(r.CRBSI*n.cathdays/1000)
)
p.CRBSI.T <- ExpressionModelVariable$new(
  'Probability of CRBSI with standard dressing', 'P',
  quote(r.CRBSI*n.cathdays*hr.CRBSI/1000)
)

# costs
c.Dermatitis.MVE <- ExpressionModelVariable$new(
  'Cost of dermatitis', 'GBP',
  quote(c.Dermatitis)
)
#> Warning in .subset2(public_bind_env, "initialize")(...):
#> ExpressionModelVariable$new: expr must be of type 'call'
c.LSI.MVE <- ExpressionModelVariable$new(
  'Cost of LSI', 'GBP',
  quote(c.LSI)
)
#> Warning in .subset2(public_bind_env, "initialize")(...):
#> ExpressionModelVariable$new: expr must be of type 'call'
c.CRBSI.MVE <- ExpressionModelVariable$new(
  'Cost of CRBSI', 'GBP',
  quote(c.CRBSI)
)
#> Warning in .subset2(public_bind_env, "initialize")(...):
#> ExpressionModelVariable$new: expr must be of type 'call'
c.S <- ExpressionModelVariable$new(
  'Cost of standard dressing', 'GBP',
  quote(n.dressings*c.Standard)
)
c.T <- ExpressionModelVariable$new(
  'Cost of Tegaderm', 'GBP',
  quote(n.dressings*c.Tegaderm)
)
```

Constructing the decision tree

The following code constructs the decision tree, node by node, based on figure 2 of Jenks *et al* (2016). In the formulation used by `rdecision`, each node is a potentially recursive structure which is allowed to have zero or more child nodes; any child nodes must have already been declared before their parent node is declared. This implies that a tree should be constructed from right to left, starting with leaf nodes which have no children (leaf nodes are synonymous with pathways in Briggs' terminology (2006)). The final node to be constructed is the node representing the decision problem.

```
# standard dressing branch
leaf.S.Dermatitis <- LeafNode$new('Dermatitis (Standard Dressing)')
leaf.S.LSI <- LeafNode$new('Local site infection (Standard Dressing)')
leaf.S.CRBSI <- LeafNode$new('CRBSI (Standard Dressing)')
leaf.S.NoComp <- LeafNode$new('No complication (Standard Dressing)')

chance.S <- ChanceNode$new(
  children = list(leaf.S.Dermatitis, leaf.S.LSI, leaf.S.CRBSI, leaf.S.NoComp),
  edgelabels = c('Dermatitis', 'Local site infection', 'CRBSI', 'No complication'),
  costs = list(c.Dermatitis.MVE, c.LSI.MVE, c.CRBSI.MVE, 0),
  p = list(p.Dermatitis.S, p.LSI.S, p.CRBSI.S, as.numeric(NA)),
  ptype = 'MV'
)

# Tegaderm dressing branch
leaf.T.Dermatitis <- LeafNode$new('Dermatitis (Tegaderm CHG)')
leaf.T.LSI <- LeafNode$new('Local site infection (Tegaderm CHG)')
leaf.T.CRBSI <- LeafNode$new('CRBSI (Tegaderm CHG)')
leaf.T.NoComp <- LeafNode$new('No complication (Tegaderm CHG)')

chance.T <- ChanceNode$new(
  children = list(leaf.T.Dermatitis, leaf.T.LSI, leaf.T.CRBSI, leaf.T.NoComp),
  edgelabels = c('Dermatitis', 'Local site infection', 'CRBSI', 'No complication'),
  costs = list(c.Dermatitis.MVE, c.LSI.MVE, c.CRBSI.MVE, 0),
  p = list(p.Dermatitis.T, p.LSI.T, p.CRBSI.T, as.numeric(NA)),
  ptype='MV'
)

# decision node
d <- DecisionNode$new(
  children = list(chance.S, chance.T),
  edgelabels = c('Standard Dressing', 'Tegaderm CHG'),
  costs = list(c.S, c.T)
)
```

In the manufacturer's model, the uncertainties in the probabilities associated with the polytomous chance nodes were modelled as independent variables. This is not recommended because there is a chance that a particular run of the PSA will yield probabilities that are outside the range [0,1]. Representing the uncertain probabilities with draws from a Dirichlet distribution is preferred. Creating a `ChanceNode` with ModelVariableExpressions is permitted, but results in a warning being issued.

Documenting the model

Package `rdecision` includes tools for automated documentation of the model structure and inputs.

Model inputs

The model variables associated with a node and its descendants can be tabulated using the method `tabulateModelVariables`. The method returns a data frame describing each variable, its description, units and uncertainty distribution. Variables inheriting from type `ModelVariable` will be included in the tabulation; regular numeric values will not be listed. For extensive models, variables associated with separate

branches of a tree can be tabulated separately by calling the method for different head nodes. Selected columns are omitted from the

```
local({
  DF <- d$tabulateModelVariables(descend=F, explode=T)
  #DF$Variable <- NULL
  #DF$SD <- NULL
  knitr::kable(DF, row.names=F, format.args=list(scientific=F), digits=3)
})
```

Description	Units	Distribution	Mean	SD	Q2.5	Q97.5
Cost of standard dressing	GBP	n.dressings * c.Standard	4.02	NA	NA	NA
No. dressings	dressings	N(3,0.3)	3.00	0.3	2.412	3.588
Standard dressing cost	GBP	Constant	1.34	0.0	1.340	1.340
Cost of Tegaderm	GBP	n.dressings * c.Tegaderm	18.63	NA	NA	NA
No. dressings	dressings	N(3,0.3)	3.00	0.3	2.412	3.588
Tegaderm CHG cost	GBP	Constant	6.21	0.0	6.210	6.210

Running the model

The following code runs a single model scenario, using the `evaluate` method of a decision node to evaluate each pathway and decision option. In the model there are eight possible root-to-leaf paths, each of which begins with the decision node and ends with a leaf node. For example, pathway `Dermatitis (Standard Dressing)` involves a traversal of nodes `d`, `chance.S`, and `leaf.S.Dermatitis`. This model did not consider utility, and the columns associated with utility are removed.

```
RES <- d$evaluate(expected=T)
RES$Utility <- NULL
RES$ExpectedUtility <- NULL
```

Model results

Base case

The results of the scenario model, using the code from the previous section, yields the following result:

Choice	Pathway	Probability	Cost	ExpectedCost
Standard Dressing	Dermatitis (Standard Dressing)	0.0078000	154.02	1.20
Standard Dressing	Local site infection (Standard Dressing)	0.1000000	254.02	25.40
Standard Dressing	CRBSI (Standard Dressing)	0.0148000	9904.02	146.58
Standard Dressing	No complication (Standard Dressing)	0.8774000	4.02	3.53
Tegaderm CHG	Dermatitis (Tegaderm CHG)	0.0387129	168.63	6.53
Tegaderm CHG	Local site infection (Tegaderm CHG)	0.0434406	268.63	11.67
Tegaderm CHG	CRBSI (Tegaderm CHG)	0.0064292	9918.63	63.77
Tegaderm CHG	No complication (Tegaderm CHG)	0.9114172	18.63	16.98

There are, as expected, eight root-to-leaf pathways. The total probability and expected cost for each choice can be calculated from the table, giving a point estimate of the saving of 77.76 GBP. This is close to the sponsor's point estimate of cost saving estimated from their probabilistic sensitivity analysis, 77.26 GBP, reported in Jenks *et al* (2016).

Probabilistic senstivity analysis

When they are created, each `ModelVariable` returns its expected value when its method `value()` is called. Calling the method `sample()` of a model variable causes it to sample from its uncertainty distribution, and return the sampled value when method `value()` is next called. The same sampled value will be returned until `sample()` is called again. Calling `sample(expected=T)` causes `value()` to return the expected value of the variable.

Probabilistic sensitivity analysis is supported through the use of sampling model variables. In practice, because the model variables are contained within model variable expressions embedded in the model, decision nodes expose a method called `sample()`. This causes each model variable associated with the decision node and its descendants to be sampled. The code needed to run N samples of the model, and the results of the first 10 samples are as follows:

```
local({
  N <- 1000
  RES <<- data.frame(
    Run = 1:N,
    Tegaderm = numeric(length=N),
    Standard = numeric(length=N)
  )
  for (i in 1:N) {
    RUN <- d$evaluate(expected=F)
    RES$Tegaderm[i] <<- sum(RUN$ExpectedCost[RUN$Choice=='Tegaderm CHG'])
    RES$Standard[i] <<- sum(RUN$ExpectedCost[RUN$Choice=='Standard Dressing'])
  }
  RES$Difference <<- RES$Tegaderm - RES$Standard
  knitr::kable(head(RES, n=10))
})
```

Run	Tegaderm	Standard	Difference
1	127.76	162.46	-34.70
2	183.31	152.08	31.23
3	119.42	203.20	-83.78
4	78.32	143.33	-65.01
5	144.57	195.99	-51.42
6	85.53	95.39	-9.86
7	101.49	195.62	-94.13
8	111.13	129.81	-18.68
9	82.55	157.67	-75.12
10	130.66	152.76	-22.10

From PSA, the mean cost of treatment with Tegaderm was 99.66, the mean cost of treatment with standard dressings was 174.87 and the mean cost saving was -75.21. The 95% confidence interval for cost saving was -150.33 to 22.75. Overall, 94.4% of runs found that Tegaderm was cost saving. These results replicate those reported by the manufucturer.

An alternative, tree-free approach

It is possible to solve the decision problem without first constructing a tree, by combining model variables directly. This is, in essence, the approach taken in many decision tree models constructed in Excel. Each component of the cost of the technology and its comparator is as follows:

```
# component costs, standard dressing
CHG.S <- ExpressionModelVariable$new(
  "Cost of standard dressing", "GBP",
  quote(n.dressings*c.Standard)
)
CRBSI.S <- ExpressionModelVariable$new(
  "Cost of CRBSI, standard dressing", "GBP",
  quote(c.CRBSI*r.CRBSI*n.cathdays/1000)
)
LSI.S <- ExpressionModelVariable$new(
  "Cost of LSI, standard dressing", "GBP",
  quote(c.LSI*r.LSI)
)
Dermatitis.S <- ExpressionModelVariable$new(
  "Cost of dermatitis, standard dressing", "GBP",
  quote(r.Dermatitis*c.Dermatitis*n.dressings)
)

# component costs, Tegaderm
CHG.T <- ExpressionModelVariable$new(
  "Cost of Tegaderm", "GBP",
  quote(n.dressings*c.Tegaderm)
)
CRBSI.T <- ExpressionModelVariable$new(
  "Cost of CRBSI, Tegaderm", "GBP",
  quote(c.CRBSI*r.CRBSI*hr.CRBSI*n.cathdays/1000)
)
LSI.T <- ExpressionModelVariable$new(
  "Cost of LSI, Tegaderm", "GBP",
  quote(c.LSI*r.LSI*hr.LSI)
)
Dermatitis.T <- ExpressionModelVariable$new(
  "Cost of dermatitis, Tegaderm", "GBP",
  quote(r.Dermatitis*c.Dermatitis*rr.Dermatitis*n.dressings)
)

# savings
c.diff <- ExpressionModelVariable$new(
  "Saving", "GBP",
  quote((CHG.T+CRBSI.T+LSI.T+Dermatitis.T) - (CHG.S+CRBSI.S+LSI.S+Dermatitis.S))
)
```

Base case

The components of cost can be extracted from the final ExpressionModelVariable, as follows:

```
#local({
#  BASE <- c.diff$tabulateModelVariables()
#  keep <- c('Description', 'Units', 'Mean')
#  knitr::kable(BASE[,keep], row.names=F, format.args=list(scientific=F), digits=2)
#})
```

The point estimate of the saving is -77.76, identical to the point estimate saving calculated using the decision

tree approach.

Probabilistic sensitivity analysis

Each model variable provides a method, `r`, to make random draws from its uncertainty distribution. The PSA for the cost difference can therefore be achieved by a single call to this method:

```
tf.psa <- c.diff$r(10000)
```

From this distribution, the mean cost saving was -77.55, the 95% confidence interval of the saving was -140.33 to -11.72 and 98.52% of runs generated a cost saving. Within fluctuation error this is consistent with the manufacturer's reported saving of 77.26 with 98.5% of runs being cost saving.

References

Briggs, Andrew, Karl Claxton, and Mark Sculpher. 2006. *Decision Modelling for Health Economic Evaluation.* Oxford, UK: Oxford University Press.

Jenks, Michelle, Joyce Craig, William Green, Neil Hewitt, Mick Arber, and Andrew J. Sims. 2016. "Tegaderm CHG IV Securement Dressing for Central Venous and Arterial Catheter Insertion Sites: A NICE Medical Technology Guidance." *Applied Health Economics and Health Policy.*