Set Cover Problem

Melissa Wong

January 13, 2020

1 Background

Once scores have been entered into the MAADCAP tool, one may want to determine the minimum set of data types that give the maximum coverage across threats. This is known as the *set cover problem*.

Given:

- Universe $U = u_1, u_2, ..., u_n$
- Subsets $S_1, S_2, ..., S_k \subseteq U$
- Costs $c_1, c_2, ..., c_k$

Goal:

- Find a set $I \subseteq \{1, 2, ...k\}$ that minimizes $\sum_{i \in I} c_i$ such that $\cup_{i \in I} S_i = U$

(Note: for the unweighted problem, all $c_i = 1$)

Finding the optimal solution is NP-hard. However, there are two approximations to the optimal solution, Linear Programming Relaxation with Randomized Rounding and the Greedy Method. Both methods are $O(\log n)$; however the Greedy Method is simpler to understand and thus will be the focus of this paper.

2 Greedy Method

The Greedy Method is an iterative algorithm that chooses the set whose cost effectiveness is smallest at each iteration. Let C represent the set of elements covered so far. Let cost effectiveness (α) be averge cost per newly covered node.

1. Set $C \leftarrow \emptyset$
2. While $C \neq U$

- Find the set S with the smallest $\alpha = \frac{c(S)}{|S-C|}$
- $C \leftarrow C \cup S$

3. Output picked sets

The Greedy Method has an upper bound of $m \log n$ where m is the number of sets in the optimal solution.

For the MAADCAP application, two modifications are necessary. First, we cannot assume that union of all subsets equals the universe (i.e., there may be a gap where a threat cannot be detected in any data source). So the stopping criteria will need to modified. Second, the weight may not be constant for a given data type across all threats. For example, if we choose analytic complexity as a proxy for weight, then a data type may have low analytic complexity (i.e., weight) for some threats and high analytic complexity for other threats. An example is illustrated below.

2.1 Example

```
# Set seed for reproducibility
set.seed(123456)

# Model Universe (corresponds to threats in MAADCAP)
U <- c("A", "B", "C", "D", "E", "F", "G", "H")
n <- length(U)

# Create an array of m x n random scores (m is number of data types, n is number of threats in MAADCAP)
# 0 - no coverage; 1 to 5 - analytic complexity
m <- 5
scores <- matrix(sample(c(0, 1, 2, 3, 4, 5), size = m*n, replace = TRUE,
                        prob = c(0.75, 0.05, 0.05, 0.05, 0.05, 0.05)),
                        nrow = m, ncol = n)
colnames(scores) <- U
rownames(scores) <- paste("set", 1:m, sep="")

scores
```

```
##      A B C D E F G H
## set1 2 0 2 4 0 0 0 0
## set2 2 0 0 4 0 4 4 1
## set3 0 0 5 0 0 4 0 4
## set4 0 1 4 0 0 4 0 0
## set5 0 0 1 2 0 0 3 0
```

```
source("../analysis/set_cover.R")

# Get set cover using scores as weights

set_cover(scores)
```

```
## $C
## [1] "C" "D" "G" "A" "F" "H" "B"
##
## $W
## C D G A F H B
## 1 2 3 2 4 1 1
##
## $set_ids
## set4 set2 set1 set5
##    4    2    1    5
```

```
# Compare to result with unweighted scores

set_cover(scores, weighted=FALSE)
```

```
## $C
## [1] "A" "D" "F" "G" "H" "B" "C"
##
## $W
## A D F G H B C
## 2 4 4 4 1 1 4
##
## $set_ids
```

```
## set4 set2
##    4    2
```