`SQM`: An R Package for Signature Quality Metrics The Utility Component

John A. Ramey

April 30, 2012

1 Introduction

The `SQM` package contains the R code for the Signature Quality Metrics project under the Signature Discovery Initiative (SDI) at the Pacific Northwest National Laboratory (PNNL). The purpose of the SQM project is to provide subject-matter experts (SMEs) with a set of tools to assess the quality of a set of signatures in terms of accuracy, cost, and utility. For each of these components, we have provided easy-to-use, accessible functions to measure the quality of inputted signatures. In this document, we discuss the `Utility` component and demonstrate its usage with an example[1].

The `Utility` component of `SQM` enables an SME to indicate his value for a set of signature system attributes via a mathematical formulation. Paired with other SQM components, such as `Cost`, the `Utility` component equips an SME to identify signature systems that are of high valuee, are cost-effective, and attain excellent accuracy.

In this document, we demonstrate the `Utility` component of the `SQM` framework with a motivating example. We have completely fabricated the example with contrived numbers and a fictitious SME, to illustrate a realistic application of the `SQM` framework.

2 Example: Three Sensor Systems

To motivate the `Utility` component, we consider a hypothetical scnenario, where we wish to compare three portable, human-carried sensor systems, which we denote as:

- Sensor `A`
- Sensor `B`
- Sensor `C`.

[1]Familarity with R is recommended to fully comprehend our provided example.

Each of the three systems can be compared with a variety of metrics, such as their accuracy in various, real-world scenarios and their cost. These two metrics can be evaulated with the `Accuracy` and `Cost` components of the `SQM` package. Additionally, we may wish to compare the sensor systems in terms of their overall usefulness, which may be difficult to compute, based on our subjective evaluation of *other attributes* of a sensor system. In this document, we consider the following two *other attributes* of the sensor systems:

1. Weight, w
2. Stand-off distance, d.

For a deployed sensor system, the system's weight must be considered in addition to its cost and accuracy. Consider, for example, a sensor that is highly accurate but cannot be carried by a single human operator, due to its excessive weight. Clearly, excessive weight is inhibiting and can reduce the sensor's practicality in the field. So, we consider weight as an additional attribute with the understanding that lighter is better.

Also, we consider the stand-off distance, which is the distance a sensor can be from its target and maintain a specificied accuracy. The larger the stand-off distance of a system is, the better. For illustration, we (naively?) assume that the range of stand-off distances has little impact on the accuracy of detection. That is, we assume that the accuracy of the sensor system practically does not change within the considered range of stand-off distances. Moreover, we are not interested in its impact on accuracy of detection. Rather, we wish to elicit from the SME the perceived value of stand-off distance in terms of additional, perhaps unknown, attributes corresponding to stand-off distance. For example, if a sensor system were used at an airport security portal to detect illegal narcotics, we may wish for the stand-off distance of the detection system to be large enough to avoid alerting the miscreant that they are being detected; however, for this example, a minimum recommended stand-off distance may not be clear, as it could be subjective.

In our example problem, each of the two *other attributes* will be used to inform the SME of the three sensor systems' *utility*. The purpose is to formalize the subjectivity of the preference of the other attributes. For instance, while we can easily say that *heavy* sensors are impractical, we wish to be upfront with what we mean by *heavy* and also to combine our assessment of weights with the stand-off distance.

To determine the SME's preference for the values of each attribute, we determine a function that maps the observed values of the attribute to the unit interval. Each attribute potentially can have its own utility (value) function if the attributes can be assumed independent or separable. The elicitation of these functions is non-trivial and can demand expertise from an expert in decision analysis. In our example, we demonstrate an approach with simplicity in mind and remark that realistic applications may require more sophisticated utility functions.

Our goal is to provide a single utility score, measured in *utils*, between 0 and 1, inclusively, that is used to rank each considered sensor system. The sensor system with the maximum aggregated utility score is said to have the highest value as defined by the elicited utility functions, which we discuss below. Furthermore, the utility scores of each system can then be matched with their respective accuracies, costs, and risks to identify the best overall sensor.

2.1 Utility Function Elicitation

To illustrate the eliciation of the utility functions for each feature, we require some notation. Suppose that we have n signatures, each with p attributes (features), that we wish to compare and determine the most valuable signature. In our example, we have $n = 3$ signatures having $p = 2$ features. In this example, we make the simplifying assumption that *weight* and *stand-off distance* are independent. We remark that this assumption could be naive in realistic applications, but for demonstration of the `SQM` package, it allows us to effectively show the usage of the utility component.

We denote by x_{ij} the jth observed feature for signature with $i = 1, \ldots, n$ and $j = 1, \ldots, p$. Then, we write the utility function for the jth feature j as $U_j(x_{ij})$. Notice that the same utility function is applied to each signature.

There are vast collections of functional forms that may be used for these utility functions. As noted above, we are aiming for simplicity. First, we standardize the value x_{ij} to obtain a value in the unit interval. We use a reasonable approach by asking the SME what range of values can the feature reasonably expected to have. For example, although the weight of a system can possibly be 2 million tons, this amount of weight is perhaps absurd to even consider because no such system would ever be considered in practice. Thus, we ask that the SME provide x_j^{min} and x_j^{max}, the minimum and maximum reasonable values for feature j, respectively. We standardize the x_{ij} by

$$x_{ij}^* = \frac{x_{ij} - x_j^{min}}{x_j^{max} - x_j^{min}},$$

which is now between 0 and 1, inclusively.

For the jth feature, we write the utility function $U_j(x_{ij})$ as the exponential function

$$\begin{aligned} U_j(x_{ij}) &= \frac{\exp\left\{-k\left(\dfrac{x_{ij} - x_j^{min}}{x_j^{max} - x_j^{min}}\right)\right\}}{\exp\{-k\} - 1} \\ &= \frac{\exp\{-kx_{ij}^*\} - 1}{\exp\{-k\} - 1}. \qquad (1) \end{aligned}$$

The exponential function in (1) allows for a variety of shapes that provide a reasonable valuation of the utility functions for our example. The exponential rate of (1) is determined by k. Also, notice that $U_j(x_j^{min}) = 0$ and $U_j(x_j^{max}) = 1$. After the SE indicates the minimum and maximum acceptable values for the jth feature, we need only determine the value of k. To elicit the value of k from the SME, we can ask for a single point that lies on the exponential curve. Rather than bombarding the SME with equations to grok, we can implicitly determine the value for k through a straightforward exercise.

2.1.1 Exercise to Determine the Shape of the Exponential Function

Rather than asking the user for a shape parameter, we can ask the user for a specified value of the jth feature and elicit a utility value through which the exponential function in (1) must pass. As before, we wish to avoid asking for functional values (much less, mentioning equations at all). We instead can provide the user with the following scenario prompt:

> The purpose of this exercise is to rate the relative the desirability of different levels of a given attribute, assuming a fixed level of the other.
>
> The least desirable level is rated 0 out of 100 points, and the most desirable level a score of 100 out of 100 points. You will be given an intermediate level to which you should assign your personal rating between 0 and 100. Your rating should reflect how valuable it is to you to move from th eleast desirable to the intermediate level relative to moving from the least to the most desirable level.
>
> If it's not worth much to you to move from the least desirable to the intermediate level, then your rating should be relatively low (e.g. less than 50 – perhaps, much less than 50). Alternatively, if moving to the intermediate level is almost as valuable as moving all the way to the most desirable level, then your rating should be relatively high (e.g. more than 50).

The functional form in (1) requires that the elicited scores be scaled down to the unit interval. We can achieve this simply by dividing the given scores by 100.

Next, we have the user provide a score for each feature with a well-crafted question. For example, suppose that the SME has indicated that an acceptable range of weights of a sensor system is between 0 and 60 pounds. The SME states that values above 60 pounds can be excessive because the professional may already be carrying a large amount of weight. We identify the midpoint of the values to be 30 pounds. Then, we ask the SME the following question:

> On the scale from 0 to 100, how would you rate a signature system weighing 30 pounds?

Our fictitious SME responds with a score of 75, based on the following reasoning. For systems that are light in weight, changes are hardly noticeable to the professional and barely influence the perceived value. However, for larger weights of deployed systems, small changes in weight are considerable and severely influence the system's perceived value.

Next, we ask the SME a similar question to elicit a value function for stand-off distance. The SME has indicated that an acceptable range of stand-off distances is 0 to 100 feet, so that the accuracy is near constant. As before, we compute the midpoint to be 50 feet and ask:

> On the scale from 0 to 100, how would you rate a signature system with a stand-off distance of 50 feet?

The SME provides the score of 30. The SME argues that small stand-off distances only have a small amount of perceived value. For these small distances, small changes in the distance have little influence on the system's perceived value. Eventually though, for a system to achieve greater and greater distances has huge implications for his project.

2.2 Utility Aggregation via the Swing Weight Method

Our goal is to aggregate the individual utility functions for each attribute to summarize the overall utility of a specified signature system. More formally, we wish to calculate the overall utility, U_i for signature system, $i = 1, \ldots, n$. For the signature system i, we wish to calculate the utility of the jth attribute, x_{ij} for each of the p attributes ($j = 1, \ldots, p$). We write the overall utility U_i for signature system i as

$$U_i = \sum_{j=1}^{p} \alpha_j U_j(x_{ij}), \tag{2}$$

with $\sum_{j=1}^{p} \alpha_j = 1$, where α_j weights the contribution of the jth attribute to the overall utility. For our contrived example, we elicit the weights, α_j, from our fictition SME via the swing weights method.

We first asked our fictitious SME the following question: "Suppose that we wish to increase each *other attribute* from its worst to best setting. Which of the *other attributes* is most imporant to shift from its worst-case to its best-case scenario?" The SME responded that a signature system simply cannot be deployed if its stand-off distance is near zero because several scenario contexts do not allow for a system to be close; for instance, if the signature system is employed during a covert operation, then the operation's success may not be possible if the system must be close to the target.

We then followed up our initial question with the following: "Suppose that we assigned a score of 100 on a scale of $0-100$ to shifting the stand-off distance from its worst to best value. Relative to stand-off distance, on a scale from $0 - 100$, what score would you assign a signature system's weight to shift it from its worst possible value to its best possible value?"

The SME commented that although its largest (i.e. worst) weight is quite impractical, it is conceivable that the system could still be deployed. However, he noted that practically the system cannot be heavy. He then assigned a score of 60 to weight.

To summarize, the SME assigned a score of 100 to stand-off distance and a score of 60 to weight. We compute α_1 for stand-off distance and α_2 for weight by:

$$\alpha_1 = \frac{60}{100+60} = 0.375$$
$$\alpha_2 = \frac{100}{100+60} = 0.625$$

3 Implementation with the `SQM` Package in `R`

In this section, we continue the above example and demonstrate how to employ the Utility component of the `SQM` framework's `R` package to obtain results necessary for determining a best sensor system. We require two sets of information from the above example within the `R` package:

1. The elicited utility function values along with the swing weights
2. The measured attributes for each sensor system

The primary interface with the Utility component is through the `afUtility` function, which requires that the two sets of information be provided in separate CSV files. Before we discuss the usage of the `afUtility` function, we provide the formatting of the CSV files that will be inputted.

First, we store the SME-elicited utility function arguments in the CSV file, **utilityAttributes.csv**. The column names should be provided, and the columns should be separated by commas. The CSV file has the following form:

```
##   varNames x0   y0 xmin xmax weights increasing
## 1   weight 30 0.75    0   60   0.375      FALSE
## 2 standOff 50 0.30    0  100   0.625       TRUE
```

The values for `x0` and `y0` are the elicited midpoints and their corresponding scores, respectively, for each attribute.

Next, we store the measured weights and stand-off distances for each sensor system (i.e. signature) in the CSV file, **signatures.csv**. The column names should be provided, and the columns should be separated by commas. The CSV file has the following form:

```
##     signatureID weight standOff
## 1 sensorSystemA    100       20
## 2 sensorSystemB     25       50
## 3 sensorSystemC     40       80
```

Notice that the Sensor System A is the heaviest at 100 pounds, while Sensor System B is the lightest at 25 pounds. Similarly, notice that Sensor System B has the farthest stand-off distance of 80. Our remaining question is: Which system is best? Clearly, Sensor System A is outperformed by Sensor Systems B and C, but the cost and accuracy of this system might be worth the additional weight and reduced stand-off distance. As we have discussed, the focus of this document is only on the Utility of the sensor systems.

Until now, we have yet to use the `SQM` package directly. We are ready to do so here and illustrate the `SQM` package. Once the SME-elicited information has been properly formatted into CSV files, as above, we only have two lines of code that we need to write. They are given here:

```
library(SQM)
utilityResults <- afUtility(signaturesFile = "signatures.csv",
utilityAttributesFile = "utilityAttributes.csv",
    outputFilePrefix = "utilityResults")
```

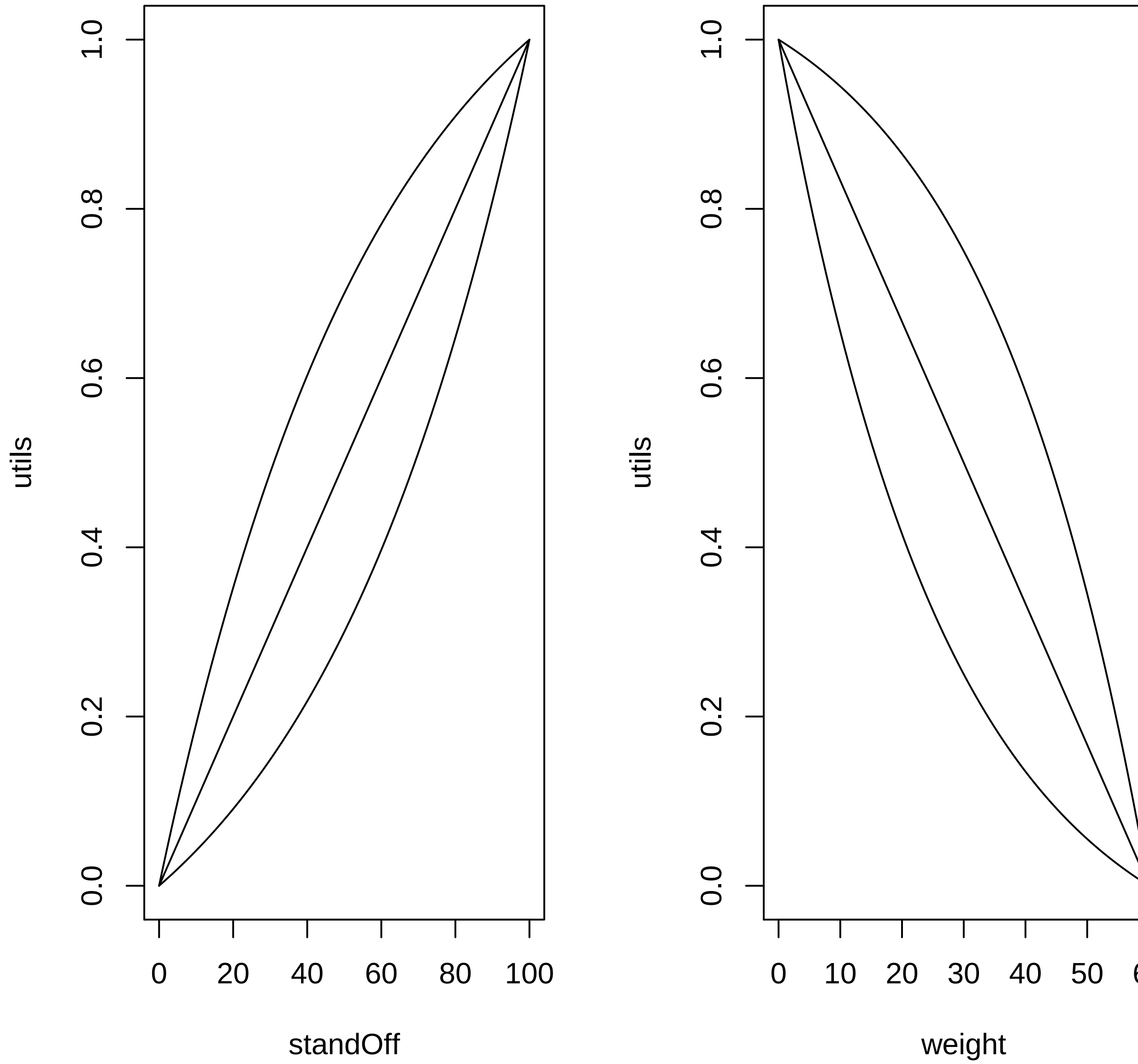
The `utilityResults` object contains useful information about the utility of each sensor system along with its attributes. For instance, we can see the fitted utility function information by attribute with

```
utilityResults$varUtilityFit

##                k xmin xmax increasing
## standOff -1.695    0  100       TRUE
## weight   -2.197    0   60      FALSE
```

To see graphs of the the elicted utility functions for each attribute, we need only call the `plotUtility` function, which uses the information in `varUtilityFit` above:

```
plotUtility(utilityResults)
```



Finally, to see the aggregated utility scores for each attribute, we can look at the `aggregate` object within the `utilityResults` object:

```
utilityResults$aggregate

##     signatureID utility
## 1 sensorSystemA -2.3051
## 2 sensorSystemB  0.6205
## 3 sensorSystemC  0.6080
```

As we can see, using the elicited utility functions, Sensor Systems B and C are greatly superior in terms of utility to Sensor System A. Sensor System B has a higher aggregated utility score than Sensor System C, but the different might not be practically different.