

Ethelo function enhancements

October 31, 2020

Abstract

We build upon the document “Calculating Ethelo” and discuss possible enhancements which can be made to the calculation of the Ethelo function, from both a theoretical standpoint as well as a computational one. This should also be taken in conjunction with the spreadsheet Calculating Ethelo Tests (spreadsheet) contained within is a computational example on how the Ethelo function is evaluated.

From the mathematical perspective, we are solving is a Mixed Integer Nonlinear Programming optimization problem. Specifically, one where all of the variables take on binary (i.e., 0 or 1) values.

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}) \\ & \text{s.t.} \\ & \vec{g}^L \leq \vec{g}(\mathbf{x}) \leq \vec{g}^U \end{aligned}$$

The scalar function $f(\mathbf{x})$ that we are minimizing is the \in function and we will define it from a functional perspective shortly. The \in -function depends on the vector \mathbf{x} (essentially \mathbf{x} defines the choices in a single scenario):

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} \quad (1)$$

where each entry x_i is a binary variable taking the value of either 0 (the option is not selected) or 1 (the option is selected). The constraint vector $\vec{g}(\mathbf{x})$ contains m potentially nonlinear functions, given by the user configuration. The objective of this optimization problem is to find the minimal feasible solution (i.e., a solution satisfying all constraint equations, and $f(\mathbf{x})$ takes the minimal value over all feasible solutions). To solve such a minimization problem, a customized version of BONMIN is used, with the following single change in the file `Bonmin/src/Algorithms/Branching/BonChooseVariable.cpp:L688-691` is:

```
// handle case where infeasible both down and up
if (results_[i].downStatus()==1 && results_[i].upStatus()==1) {
    continue;
}
```

1 The Ethelo function

From the users point of view, they input their preferences about a number of options (the options number n in total). These are known as their *Option Scores*, in the range $[-1, 1]$. Suppose that N users rank their preferences, this means that the k th user’s preferences can be encoded in a (row) vector \mathbf{p}_k^T , which can thus

be incorporated into an $N \times n$ matrix \mathbf{P} :

$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1^T \\ \vdots \\ \mathbf{p}_N^T \end{bmatrix}. \quad (2)$$

We call \mathbf{P} the matrix of influents.

1.1 Satisfaction

Given a specific value for the scenario (i.e., the dependent variable \mathbf{x}), we can compute user k 's satisfaction $\vec{\rho}_k$ with this outcome via the dot product of their preference vector with the decision, that is $\vec{\rho}_k = \mathbf{p}_k^T \mathbf{x}$. To calculate all N user's satisfaction with the outcome \mathbf{x} , we simply write this as the matrix-vector product

$$\vec{\rho} = \mathbf{P}\mathbf{x}. \quad (3)$$

1.2 Support \approx Influence = I

The ‘‘Support’’ or ‘‘Influence’’ of an outcome is the average of the score that people give for an outcome. That is, we take an average of the satisfaction:

$$I = \frac{1}{N} \sum_{k=1}^N \vec{\rho}_k. \quad (4)$$

This may be represented in vector form (and will be used later) as

$$I = \frac{1}{N} \mathbf{1}^T \vec{\rho} \quad (5)$$

$$= \frac{1}{N} \mathbf{1}^T \mathbf{P}\mathbf{x}. \quad (6)$$

Where $\mathbf{1}$ is the N -vector containing the value 1 in each element.

1.3 Dissonance = DS

The ‘‘Dissonance’’ measures the variability of the user's responses for a particular outcome. This is also known as the variance of the satisfaction. The Dissonance is defined as

$$DS = \left[\frac{1}{N} \sum_{k=1}^N \vec{\rho}_k^2 \right] - \left[\frac{1}{N} \sum_{k=1}^N \rho_k \right]^2, \quad (7)$$

and as before, we represent this in matrix form as

$$DS = \frac{1}{N} \vec{\rho}^T \vec{\rho} - \frac{1}{N^2} (\mathbf{1}^T \mathbf{P}\mathbf{x})^2 \quad (8)$$

$$= \frac{1}{N} \mathbf{x}^T \mathbf{P}^T \mathbf{P} \mathbf{x} - \frac{1}{N^2} (\mathbf{1}^T \mathbf{P}\mathbf{x})^2. \quad (9)$$

Note also that the Dissonance is always a positive number (it is the variance, after all).

The Dissonance can be represented in matrix form (notice that the two scalars in the second term may be transposed without changing their value) as

$$DS = \frac{1}{N} \mathbf{x}^T \mathbf{P}^T \left(\mathbf{I} - \frac{1}{N} \mathbf{1} \mathbf{1}^T \right) \mathbf{P} \mathbf{x}, \quad (10)$$

where \mathbf{I} is the identity matrix, and $\mathbf{1}$ is an N -dimensional vector with all entries equal to 1. Defining the matrix \mathbf{Q} as

$$\mathbf{Q} = \frac{1}{N} \mathbf{P}^T \left(\mathbf{I} - \frac{1}{N} \mathbf{1} \mathbf{1}^T \right) \mathbf{P}, \quad (11)$$

we clearly see the quadratic nature of the Dissonance term:

$$DS = \mathbf{x}^T \mathbf{Q} \mathbf{x}. \quad (12)$$

1.4 Fairness \approx Unity = U

Unity is a measure of the internal cohesion of a group with respect to an option or outcome, due to agreement or fairness in the distribution of support. Unity is referred to as U in the equations. U can range from negative, when a group is polarized in its support, to positive when a group is unified. We take the tipping point “ t ” as the neutral mid-point where a group is neither polarized nor unified; i.e. $U(t) = 0$.

1.4.1 Properties of Unity

The unity function is defined as: some function mapping DS onto a range $\{-1, 1\}$, where $U(1) = -1$, $U(0) = 1$, and $U(t) = 0$, t is the “tipping point”.

1.4.2 Current Unity function

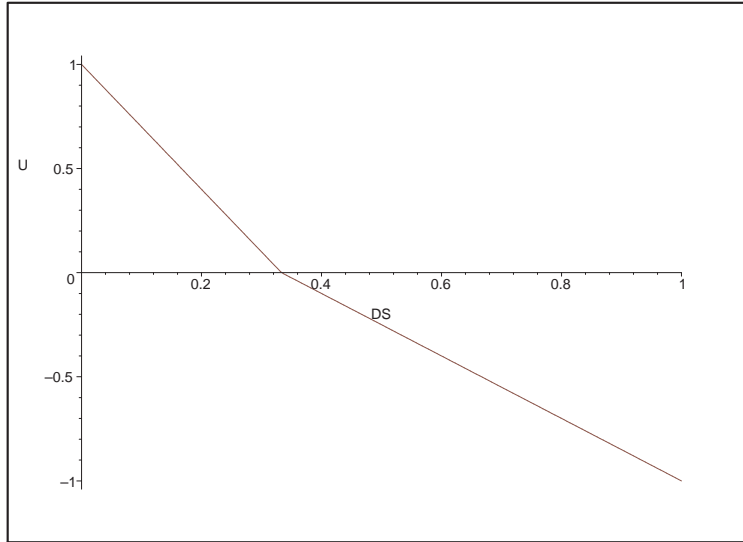


Figure 1: Current Unity as a function of Dissonance

The current Unity function is defined as a piecewise linear function passing through the three points described. We may write the unity function as

$$U(DS) = \begin{cases} \frac{(t - DS)}{t} & \text{if } DS < t \\ \frac{(t - DS)}{(1 - t)} & \text{if } t \leq DS \end{cases} \quad (13)$$

1.5 Collective Identity

The Collective Identity Ξ parameter determines the weight of Unity on a decision. This is essentially a scaling factor of the unity function.

1.6 Ethelo function

The Ethelo function \in has different definitions depending on the value of the support I and Dissonance DS values, there are four cases to consider

$$\in = \begin{cases} I + \Xi \cdot U(DS) \cdot (1 - I) & \text{for } DS < t, I > 0 \\ I + \Xi \cdot U(DS) \cdot I & \text{for } t < DS, I > 0 \\ I + \Xi \cdot U(DS) \cdot (-1 - I) & \text{for } DS < t, I < 0 \\ I + \Xi \cdot U(DS) \cdot I & \text{for } t < DS, I < 0 \end{cases} \quad (14)$$

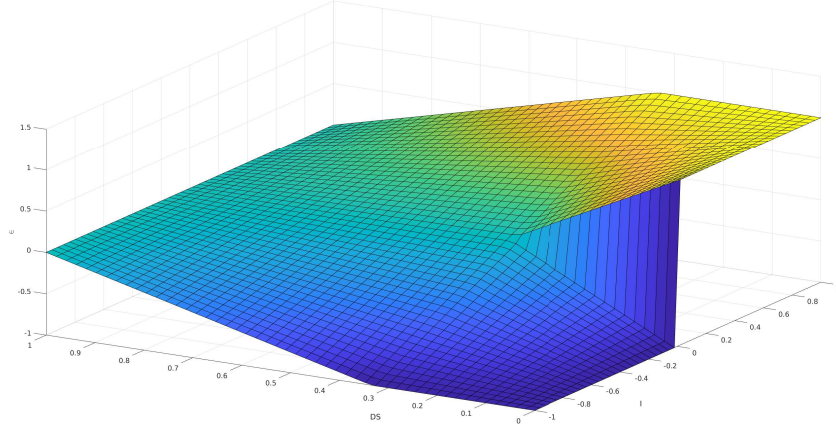


Figure 2: \in as a function of DS and I with $\Xi = 1$ and $t = 1/3$

The discontinuity in the \in function is apparent in two areas:

1. Along the line segment $I = 0$ with $DS < t$;
2. Along the line defined by $DS = t$, $I \in [-1, 1]$;

We expect that these discontinuities may present some

2 Gradient and Hessian of the \in function

From the definition of the \in function in (14), we explore the continuity of the function with respect to its parameters I and DS . Now, roughly speaking, these parameters correspond to the mean satisfaction, and its variance, so strictly speaking, they are interrelated. Indeed it is clear that they are continuous functions of the variable \mathbf{x} . What we really are interested in is the gradient of \in with respect to the variables \mathbf{x} , which we now formulate. We first expand the \in function in terms of the variables \mathbf{x} :

$$\in = \begin{cases} \frac{1}{N} \mathbb{1}^T \mathbf{P} \mathbf{x} + \frac{\Xi}{t} (t - \mathbf{x}^T \mathbf{Q} \mathbf{x}) \left(1 - \frac{\mathbb{1}^T \mathbf{P} \mathbf{x}}{N} \right) & \text{for } DS < t, I > 0 \\ \frac{1}{N} \mathbb{1}^T \mathbf{P} \mathbf{x} + \frac{\Xi}{(1-t)} (t - \mathbf{x}^T \mathbf{Q} \mathbf{x}) \frac{\mathbb{1}^T \mathbf{P} \mathbf{x}}{N} & \text{for } t < DS, I > 0 \\ \frac{1}{N} \mathbb{1}^T \mathbf{P} \mathbf{x} + \frac{\Xi}{t} (t - \mathbf{x}^T \mathbf{Q} \mathbf{x}) \left(-1 - \frac{\mathbb{1}^T \mathbf{P} \mathbf{x}}{N} \right) & \text{for } DS < t, I < 0 \\ \frac{1}{N} \mathbb{1}^T \mathbf{P} \mathbf{x} + \frac{\Xi}{(1-t)} (t - \mathbf{x}^T \mathbf{Q} \mathbf{x}) \frac{\mathbb{1}^T \mathbf{P} \mathbf{x}}{N} & \text{for } t < DS, I < 0 \end{cases} \quad (15)$$

2.1 Gradient of \in

Taking the gradient with respect to \mathbf{x} yields:

$$\nabla \in = \begin{cases} \frac{1}{N} \mathbf{P}^T \mathbb{1} - \frac{\Xi}{t} \left[2\mathbf{Q}\mathbf{x} \cdot \left(1 - \frac{\mathbb{1}^T \mathbf{P}\mathbf{x}}{N} \right) + (t - \mathbf{x}^T \mathbf{Q}\mathbf{x}) \cdot \frac{\mathbf{P}^T \mathbb{1}}{N} \right] & \text{for } DS < t, I > 0 \\ \frac{1}{N} \mathbf{P}^T \mathbb{1} - \frac{\Xi}{(1-t)} \left[2\mathbf{Q}\mathbf{x} \cdot \frac{\mathbb{1}^T \mathbf{P}\mathbf{x}}{N} - (t - \mathbf{x}^T \mathbf{Q}\mathbf{x}) \cdot \frac{\mathbf{P}^T \mathbb{1}}{N} \right] & \text{for } t < DS, I > 0 \\ \frac{1}{N} \mathbf{P}^T \mathbb{1} - \frac{\Xi}{t} \left[2\mathbf{Q}\mathbf{x} \cdot \left(-1 - \frac{\mathbb{1}^T \mathbf{P}\mathbf{x}}{N} \right) + (t - \mathbf{x}^T \mathbf{Q}\mathbf{x}) \cdot \frac{\mathbf{P}^T \mathbb{1}}{N} \right] & \text{for } DS < t, I < 0 \\ \frac{1}{N} \mathbf{P}^T \mathbb{1} - \frac{\Xi}{(1-t)} \left[2\mathbf{Q}\mathbf{x} \cdot \frac{\mathbb{1}^T \mathbf{P}\mathbf{x}}{N} - (t - \mathbf{x}^T \mathbf{Q}\mathbf{x}) \cdot \frac{\mathbf{P}^T \mathbb{1}}{N} \right] & \text{for } t < DS, I < 0 \end{cases} \quad (16)$$

Now, let us identify the quantity $\bar{\mu} = \mathbf{P}^T \mathbb{1} / N$, this is equal to the average of all of the users preferences, that is

$$\bar{\mu} = \frac{1}{N} \mathbf{P}^T \mathbb{1} = \frac{1}{N} \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \cdots & \mathbf{p}_N \end{bmatrix} \mathbb{1} = \frac{1}{N} \sum_{k=1}^N \mathbf{p}_k. \quad (17)$$

We may thus rewrite the gradient in terms of I , DS , and $\bar{\mu}$ as

$$\nabla \in = \begin{cases} \bar{\mu} - \frac{\Xi}{t} [2\mathbf{Q}\mathbf{x} \cdot (1 - I) + (t - DS) \cdot \bar{\mu}] & \text{for } DS < t, I > 0 \\ \bar{\mu} - \frac{\Xi}{(1-t)} [2\mathbf{Q}\mathbf{x} \cdot I - (t - DS) \cdot \bar{\mu}] & \text{for } t < DS, I > 0 \\ \bar{\mu} - \frac{\Xi}{t} [2\mathbf{Q}\mathbf{x} \cdot (-1 - I) + (t - DS) \cdot \bar{\mu}] & \text{for } DS < t, I < 0 \\ \bar{\mu} - \frac{\Xi}{(1-t)} [2\mathbf{Q}\mathbf{x} \cdot I - (t - DS) \cdot \bar{\mu}] & \text{for } t < DS, I < 0 \end{cases} \quad (18)$$

we may also choose to combine the second and fourth cases of this equation, since there is no discontinuity when I changes sign (indeed, this could already be done in the functional definition of the \in -function).

2.2 Hessian of \in

Taking the Jacobian of $\nabla \in$ gives us the Hessian:

$$\mathcal{H}_{\mathbf{x}}(\in) = \begin{cases} -\frac{\Xi}{t} \left[2\mathbf{Q} \cdot (1 - I) - \frac{2}{N} \mathbf{Q}\mathbf{x}\mathbb{1}^T \mathbf{P} - \frac{2}{N} \mathbf{P}^T \mathbb{1} \mathbf{x}^T \mathbf{Q} \right] & \text{for } DS < t, I > 0 \\ -\frac{\Xi}{(1-t)} \left[2\mathbf{Q} \cdot I - \frac{2}{N} \mathbf{Q}\mathbf{x}\mathbb{1}^T \mathbf{P} + \frac{2}{N} \mathbf{P}^T \mathbb{1} \mathbf{x}^T \mathbf{Q} \right] & \text{for } t < DS, I > 0 \\ -\frac{\Xi}{t} \left[2\mathbf{Q} \cdot (-1 - I) - \frac{2}{N} \mathbf{Q}\mathbf{x}\mathbb{1}^T \mathbf{P} - \frac{2}{N} \mathbf{P}^T \mathbb{1} \mathbf{x}^T \mathbf{Q} \right] & \text{for } DS < t, I < 0 \\ -\frac{\Xi}{(1-t)} \left[2\mathbf{Q} \cdot I - \frac{2}{N} \mathbf{Q}\mathbf{x}\mathbb{1}^T \mathbf{P} + \frac{2}{N} \mathbf{P}^T \mathbb{1} \mathbf{x}^T \mathbf{Q} \right] & \text{for } t < DS, I < 0 \end{cases} \quad (19)$$

3 Calculating the top choice: when there is a single influent

We can specialize the case where $N = 1$, that is, when there is only a single row in the influent matrix. We first look at the dissonance:

$$DS = \frac{1}{1} \vec{\rho}_1^2 - \frac{1}{1^2} \vec{\rho}_1^2 = 0, \quad (20)$$

this is not surprising since it is the variance, and there is only one single respondent. Furthermore, the unity function, by definition is $U(0) = 1$, and thus, our simplified \in -function is defined as:

$$\in = \begin{cases} \mathbf{p}_1^T \mathbf{x} + \Xi(1 - \mathbf{p}_1^T \mathbf{x}) & \text{for } I > 0 \\ \mathbf{p}_1^T \mathbf{x} + \Xi(-1 - \mathbf{p}_1^T \mathbf{x}) & \text{for } I < 0 \end{cases}. \quad (21)$$

In the actual functional implementation of the \in function, the collective identity is essentially set to zero (see, for example, `evaluate.cpp#L98`), and thus, the \in function reduces to:

$$\in = \mathbf{p}_1^T \mathbf{x}. \quad (22)$$

Thus, for the single influent case, this boils down to a constrained linear program. Since the decision variable \mathbf{x} is binary, the unconstrained solution to this linear program should be equal to

$$\mathbf{x} = \max(\vec{0}, \text{sign}(\mathbf{p}_1)), \quad (23)$$

and this should also be a good starting point for any further method for finding a feasible solution. The final solution will now depend mainly on the satisfiability of the constraint equations.

A Alternative Unity functions

A.1 Quadratic interpolant

Within the class of functions satisfying the properties in 1.4.1, we may choose any function interpolating those 3 values. The most obvious choice is a quadratic function (i.e., the 3 values uniquely define a quadratic curve passing through them), and this is given by

$$U(DS) = \frac{(DS - t)(DS - 1)}{t} - \frac{DS(DS - t)}{(1 - t)}. \quad (24)$$

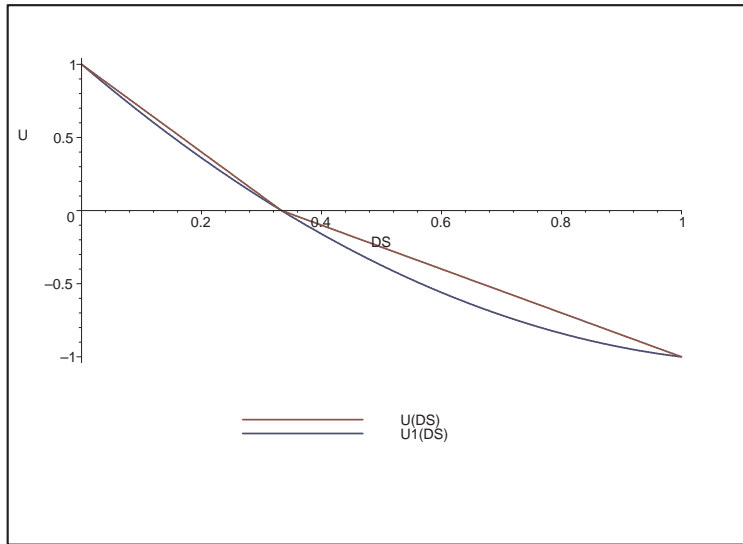


Figure 3: Proposed alternative Unity function

We show the two Unity functions together in Figure 3, and we should be able to provide a proof of the maximum deviation of the proposed Unity function based on the value of the tipping point. However, also not that for certain choices of t it is possible that the intermediate values of $U(DS)$ will fall outside the range $\{-1, 1\}$. Thus, we need to impose a restriction on the range of the “tipping point” t , to

$$t \in \left[1 - \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}} \right] \approx [0.3, 0.7]. \quad (25)$$

We could take this one step further and use a cubic interpolating spline that would allow us to impose certain conditions on the end points of the function (i.e. so that we can control the rate at which the transition between polarization and united happens, but we will first define the Ethelo function further so that we are sure that it is a good idea (remember that the underlying variable is the vector \mathbf{x} and DS is a function of this, thus we may actually want to stick with the simplest function after all).

A.2 Other nonlinear functions

Another possibility would be to use another nonlinear function to describe the change in the Unity with respect to the dissonance. One family of curves that come to mind are those that are sigmoidal in shape. There are many possibilities, one specific choice is the logistic function,

$$U_2(DS) = 1 - \frac{2}{1 + \exp -15(x - t)} \quad (26)$$

and there are, of course, myriad other possibilities in the function choice. However, we will discuss the implications of the choice at a later point.

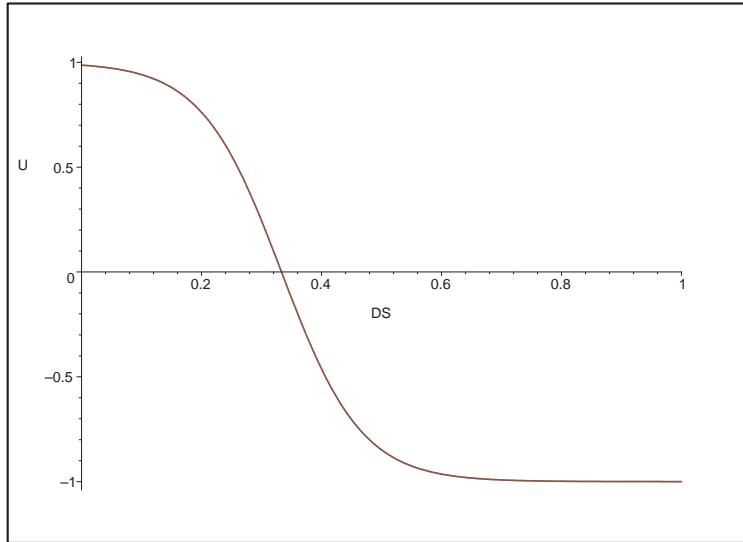


Figure 4: Proposed alternative Logistic Unity function

A Two things to keep in mind

From John Richardson (slack #engine 6-July-2020).

1. While its critical to speed up how the engine handles large groups, let's not lose sight of the importance of speed for calculating top choice—when there is a single influent. Its performing pretty well at the moment—under a few seconds for most calls—and hopefully your latest improvements will see that go down as well. I wouldn't want to see the time for single-influent calls go up as a result of these optimizations. I doubt that would be the case, but wanted to flag it just in case.
2. We spoke about this briefly, but it was early days and now that you've had a chance to review the code and applications it may land differently. In many budgeting processes, we are using "slider topic"—that is, sets of options in XOR relation to each other, that are identical except for a few attributes such as cost, and where the participants selects one of them by moving the slider to a selected "stop" or option. In the admin panel, we can choose the "base" of the triangle, whereby the engine inputs lower levels of support to options that are close to the selected stop—basically in a triangle shape. For example, say there are 10 options, each with a slightly different cost, the engine will understand the "selected options" as having 100% support, the next-door neighbours on the line to have slightly less support, etc in a triangle shape. ANOTHER APPROACH here if you can see on the back-end that this is what is happening, is to treat this as a SINGLE option, with one or more continuously variable attributes. This *may* be easier or faster to compute as a single continuous equation. The outputted results will still have to conform to the 10 option—we can't produce options that aren't in the list of 10—so there would have to be a rounding stage perhaps at the end—but that is a simple step. I'm not asking you to do this BTW! I suspect it may be complicated to implement, And I don't want to distract you from your current strategy, which may yield exactly what we want. But I want to flag it, so you understand it and can consider it with all the other things you are considering.