Decision Making in Manufacturing and Services
Vol. 8 • 2014 • No. 1–2 • pp. 5–12

Maximization of an Asymmetric Utility Function by the Least Squares

Kiyoshi Yoneda[*], Antonio Carlos Moretti[**]

Abstract. This note points out that a utility maximization procedure proposed in an earlier paper may be reduced to the least squares. The utility function is asymmetric in the sense that for each cue its ideal value and the permissible range are assigned in such a way that the ideal is not necessarily at the center of the range, like "a beer of 350 ml would be ideal, but acceptable if within [100, 500]". A practical consequence of the observation is that very little programming will be needed to deploy the utility maximization since software for the least squares is widely available.

Keywords: individual behavior, inverse problems, simultaneous equations, optimization

Mathematics Subject Classification: 90B50

JEL Classification: C44

Revised: March 17, 2014

1. INTRODUCTION

The key observation in this note is that the utility maximization proposed in Yoneda and Celaschi (2013) is isomorphic to the least squares. To explain this, we begin with a brief review of that paper.

1.1. THE UTILITY MAXIMIZATION

The aim has been to devise a tool for decision making such that anyone with a rudimentary understanding of linear equations would be able to build a model as an inverse problem and solve it by maximizing a utility function.

Consider the approximate linear equations $\phi x \approx y$, where x is a vector of decision variables, y the desired outcomes, and ϕ a matrix describing the linear causality relationship. Here, $\dim x \leqslant \dim y$ is secured by including all approximate equations of the form $x_j \approx y_j$. For concreteness, we consider an example worked out in Section 5 of Yoneda and Celaschi (2013).

[*] Fukuoka University, Faculty of Economics, Jounan-ku, Fukuoka, 814-0180 Japan, e-mail: yoneda@econ.fukuoka-u.ac.jp, corresponding author
[**] School of Applied Sciences State University of Campinas–SP, Brazil, e-mail: antonio.moretti@fca.unicamp.br

Example
A person wishes to eat peanuts and drink beer minding cost and energy gain. The requirement specifications are:

Peanuts should ideally be 50 g,
hopefully between 30 and 100.
Beer should ideally be 350 ml,
hopefully between 100 and 500.
Cost should ideally be $\frac{3}{2}$ cu,
hopefully between 0 and 5.
Energy should ideally be 200 kcal,
hopefully between 150 and 300.
and we know the formulas for cost and energy in terms of peanuts and beer.

The proposed solution method for this problem has been to maximize a utility function with respect to the decision variables. The utility function U is defined as a weighted sum of subutilities

$$\hat{x} := \arg\max \mathrm{U}(x) \qquad\qquad \mathrm{U}(x) := \sum_i w_i\, \mathrm{u}(x_i) \tag{1}$$

where u is two-piece quadratic and hence asymmetric in general, as will be detailed in Section 2.

Now, for scalar x define the subutility function $\mathrm{u}(x) := \mathrm{u}(x; y, a, b)$ with $y \in [a, b]$ satisfying the following requirements: u is continuously differentiable; u is quadratic in $[a, y]$ and also in $[y, b]$; u is linear for $x < a$ and $b < x$; and $\mathrm{u}(y) = 1$, $(d\mathrm{u}(x)/dx)(y) = 0$, $\mathrm{u}(a) = \mathrm{u}(b) = 0$. Such u exists uniquely.

$$\mathrm{u}(x; y, a, b) := \begin{cases} \dfrac{(c-x)(c-2y+x)}{(c-y)^2} & x \in [a, b] \\[2ex] \dfrac{2(c-x)}{c-y} & x \notin [a, b] \end{cases} \qquad \text{where} \qquad c := \begin{cases} a & x \leqslant y \\[2ex] b & y < x \end{cases} \tag{2}$$

This subutility function has three intuitive parameters: y, the ideal value of x; a, the lowest acceptable value of x; and b, the highest acceptable value of x. The method differs from the weighted least squares in that u is generally asymmetric around y: not necessarily $y - a = b - y$.

Example (continued)
The next expression summarizes the problem p:

$$\text{Intercept} + \phi\, x \overset{w}{\approx} y \qquad\qquad [\text{unit} \quad a \quad b \quad w]$$

$$\begin{bmatrix} 0 \\ 0 \\ 1.5 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ .02 & .01 \\ 5.92 & .406 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \overset{w}{\approx} \begin{bmatrix} 50 \\ 350 \\ 1.5 \\ 200 \end{bmatrix} \qquad \begin{bmatrix} \text{g} & 30 & 100 & .3 \\ \text{ml} & 100 & 500 & .3 \\ \text{cu} & 0 & 5 & .1 \\ \text{kcal} & 150 & 300 & .3 \end{bmatrix}$$

where:

g	:= grams	ml	:= mililiters
cu	:= currency units	kcal	:= kilocalories

which is equation (9) in Yoneda and Celaschi (2013), with the intercept left in for clarity. This system of approximate equations was solved by minimizing (1) with (2), yielding the solution:

$$\hat{x} = \begin{bmatrix} 34\ [\text{g}] \\ 150\ [\text{ml}] \end{bmatrix} \tag{3}$$

The optimization procedure permits gradient methods since u is continuously differentiable. The tails of u have been chosen to be linear rather than quadratic in order to prevent numerical overflow in resource-constrained computation devoid of the floating point arithmetics.

1.2. WHY LEAST SQUARES?

This paper points out that there is an isomorphism between the utility maximization problem described above and the least squares. The isomorphism is a bijection such that preserves the optimal solution. Its practical consequence is that very little programming will be necessary to deploy the utility maximization provided that the least squares software is readily available. The least squares programs are in fact available in various computer languages and packages, including those for hardwares with low computational capabilities such as embedded systems.

A common strategy in solving a problem p is to reduce it to an easier problem q. When q is solved, its answer r is translated back into the solution s of p. “Easier” here means that an effective tool to produce an answer is at hand.

Let:

p := problem formulated as utility maximization
q := problem formulated as loss minimization
r := solution of the loss minimization
s := solution of the utility maximization

Many computer programs exist to solve the least squares problem q which minimizes the weighted quadratic loss function

$$\hat{z} := \arg\min \mathrm{L}(z) \qquad\qquad \mathrm{L}(z) := \sum_i w_i\ \ell(z_i) \tag{4}$$

where ℓ is quadratic and hence symmetric. This method yields a solution r. The least squares method is among the best-researched topic in numerical computation.

We wish to find a pair of transformations (τ, υ) such that makes this diagram commute:

$$\begin{tikzcd} p \arrow[r, "\text{U max}"] \arrow[d, "\tau"'] & s \\ q \arrow[r, "\text{L min}"'] & r \arrow[u, "\upsilon"'] \end{tikzcd} \tag{5}$$

In the language of category theory Turi (2001) this amounts to defining an adjoint pair of functors between two categories, which in our case is merely an isomorphism $\upsilon = \tau^{-1}$, so that this diagram commutes as well:

$$\begin{tikzcd} p \arrow[r, "\text{U max}"] & s \arrow[d, "\tau"] \\ q \arrow[r, "\text{L min}"'] \arrow[u, "\upsilon"] & r \end{tikzcd}$$

The remainder of this note is organized as follows. The conversion (τ, τ^{-1}) is presented in Section 2. Section 3 considers the usefulness of this approach. Section 4 concludes the note by pointing out a direction for the future research.

2. OBSERVATION

The proposed subutility function has been (2). Its main part is for $x \in [a, b]$, which is a two-piece quadratic, and the parts for $x \notin [a, b]$ are just the linear extensions of the former. In the remainder of this section we consider only the domain $[a, b]$ until we recapture $]-\infty, \infty[$ at the end of the section.

Now, consider the standardization:

$$\tau : x \mapsto z := \frac{|x-y|}{\mathrm{c}(x) - y} \quad \text{where} \quad \mathrm{c}(x) := \begin{cases} a & x < y \\ b & y < x \end{cases} \tag{6}$$

so that $[a\ y\ b] \mapsto [-1\ 0\ 1]$. This is a two-piece linear function. The usual quadratic loss function is:

$$\ell(z) := z^2$$

Then, the subutility function is:

$$\begin{aligned} 1 - \ell(z) = 1 - z^2 = 1 - \left(\frac{x-y}{\mathrm{c}(x)-y}\right)^2 &= \frac{\{\mathrm{c}(x) - x\}\{\mathrm{c}(x) - 2y + x\}}{\{\mathrm{c}(x) - y\}^2} \\ = \mathrm{u}(x) \quad x \in [a, b] \end{aligned}$$

This means that the maximization of subutility function $\mathrm{u}(x)$ is equivalent to the minimization of the loss function $\ell(z) = z^2$, and that the value of x may be recovered by the destandardization:

$$\tau^{-1} : z \mapsto x = y + \{\mathrm{d}(z) - y\}|z| \quad \text{where} \quad \mathrm{d}(z) := \begin{cases} a & z < 0 \\ b & 0 < z \end{cases} \tag{7}$$

This is also a two-piece linear function. Now, it is obvious that the pair (τ, τ^{-1}) makes the diagram (5) commute for values $x \in [a, b] \;\Leftrightarrow\; z \in [-1, 1]$.

To summarize, for $x \in [a, b]$:

$$\begin{tikzcd}
p \arrow[d, shift right, "\tau"'] & x \arrow[r, mapsto, "\mathrm{u}", "\text{2-piece quadratic}"'] \arrow[d, mapsto] & \mathrm{u}(x) \\
q \arrow[u, shift right, "\tau^{-1}"'] & \mathrm{z}(x) \arrow[r, mapsto, "\ell", "\text{quadratic}"'] & \ell(z(x)) \arrow[u, mapsto]
\end{tikzcd}$$

Note that it is typically the case that some elements of x are linear transformations of other elements of x. In this case, (τ, τ^{-1}) is piecewise linear: see ϕ in the example. If on the other hand, the transformation is monotonic but not necessarily linear, so is (τ, τ^{-1}) with nondifferentiable points. In Section 3 we give further consideration to this point.

For $1 - \ell(z)$ to match $\mathrm{u}(x)$ in $]-\infty, \infty[$ rather than only in $[a, b]$ define:

$$\ell(z) := \begin{cases} -2z - 1 & z \in]-\infty, -1] \\ z^2 & z \in [-1, 1] \\ 2z - 1 & z \in [1, \infty[\end{cases}$$

Under this definition, $\ell(z) = 2\,|z|$ for large $|z|$, so that overflow is predictable, making computation easier for processors without floating point arithmetics.

Example (continued)
We try the reduction method. By τ the peanuts and beer problem transforms to q:

$$\varphi(z) \overset{w}{\approx} 0$$

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} \overset{w}{\approx} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad\qquad w = \begin{bmatrix} .3 \\ .3 \\ .1 \\ .3 \end{bmatrix}$$

with:

$$\mathrm{d}_1(z_1) := \begin{cases} 30 & z_1 < 0 \\ 100 & 0 < z_1 \end{cases} \qquad\qquad \mathrm{d}_2(z_2) := \begin{cases} 100 & z_2 < 0 \\ 500 & 0 < z_2 \end{cases}$$

$$x_1 = 50 + \{\mathrm{d}_1(z_1) - 50\}\,|z_1| \qquad x_2 = 350 + \{\mathrm{d}_2(z_2) - 350\}\,|z_2|$$
$$x_3 = 1.5 + .2\,x_1 + .01\,x_2 \qquad x_4 = 5.92\,x_1 + .406\,x_2$$

$$\mathrm{c}_3(x_3) := \begin{cases} 0 & x_3 < 1.5 \\ 5 & 1.5 < x_3 \end{cases} \qquad \mathrm{c}_4(x_4) := \begin{cases} 150 & x_4 < 200 \\ 300 & 200 < x_4 \end{cases}$$

$$z_3 = \frac{|x_3|}{\mathrm{c}_3(x_3)} \qquad z_4 = \frac{|x_4 - 200|}{\mathrm{c}_4(x_4) - 200}$$

Minimizing (4) yields the optimum solution r

$$\hat{z} = \begin{bmatrix} -0.783 \\ -0.800 \end{bmatrix}$$

which by τ^{-1} translates back to s, (3).

3. DISCUSSION

3.1. IMPORTANCE WEIGHTS

A clear benefit in standardization τ comes from the psychology of weight assignment Hastie *et al.* (2001):

> If one type of information (e.g., test scores) is conveyed by numbers that range from 200 to 800 and another type (e.g., grades) is conveyed by numbers that range from 1 to 4, the human brain will be fooled into greater judgment adjustments based on the "larger quantities" on the first scale. The implication is that, when intuitive judgments are made, it's good practice to standardize the cue information scales.

When the cues are presented in distinct measurement units, it is more difficult to compare their relative importances than when they are all presented in a unit-free scale in $]-1, 1[$. Therefore, it is recommendable to assign the weights w upon standardization. Although this objective may be achieved simply by standardization rather than by solving the entire problem by conversion, the latter possibility provides a theoretical justification of the standardization procedure.

3.2. DESCRIPTION AND OPTIMIZATION

Suppose the causality relationship ϕ is linear. Then, since z is a two-piece linear function of x, it is possible to write q as a linear rather than nonlinear least squares problem by introducing dummy variables. This procedure could be written in the form of a program, but the task is not simple on top of increasing the problem size. Hence, it seems labor-saving to use a nonlinear least squares software `nls` to solve q.

Suppose further that `nls` is implemented in two parts, $\mathtt{des}_q$ for description of nonlinear least squares problem and `opt` for loss function minimization. If `opt` is written in a generic way so that it can optimize not only quadratic loss functions but just about any reasonable function, which is often the case with large packages, then U can be maximized directly using `opt`.

$$\begin{tikzcd}
p \arrow[r, "\mathtt{des}_p"] \arrow[d, "\tau"'] & \mathrm{U} \arrow[r, "\mathtt{opt}"] & s \\
q \arrow[r, "\mathtt{des}_q"] \arrow[rr, bend right, "\mathtt{nls}"'] & \mathrm{L} \arrow[r, "\mathtt{opt}"] & r \arrow[u, "\tau^{-1}"']
\end{tikzcd}$$

In this case it makes sense to reduce p to q only if the labor to program both (τ, τ^{-1}) and $\mathtt{des}_q$ is less than that to program the p descriptor $\mathtt{des}_p$, a condition which seems hard to hold.

Summarizing, the problem conversion approach is useful for setting the weights w, while actually solving $p \to s$ via $p \xrightarrow{\tau} q \to r \xrightarrow{\tau^{-1}} s$ is useful only when

$$\mathtt{nls} : q \to r$$

is available in such a way that `opt` is difficult to deploy apart from `nls`.

Is this a common situation? It seems so. For instance, Python[1] has a module called `lmfit`[2], which plays the role of $\mathtt{des}_q$ above, *assuming the nonlinear least squares* for the optimization `opt` = `leastsq`. What $\mathtt{des}_q$ = `lmfit` does is facilitate model modification experiments, such as fixing some of the variables or modifying dependencies among them. With `lmfit`, exploration of possible models becomes easier.

The task for `opt` is delegated to a function called `scipy.optimize.leastsq` which connects to the Levenberg-Marquardt algorithm[3] in `MINPACK-1`[4].

$$\begin{tikzcd}
p \arrow[rr, dotted] \arrow[d, "\tau"'] & & s \\
q \arrow[r, "\mathtt{des}_q", "\mathtt{lmfit}"'] & \mathrm{L} \arrow[r, "\mathtt{nls}", "\mathtt{leastsq}"'] & r \arrow[u, "\tau^{-1}"']
\end{tikzcd}$$

4. CONCLUSION

The advantage of decomposing the utility maximization into the problem description and the least squares seems clear. Unfortunately, the description framework provided by $\mathtt{des}_q$ = `lmfit` is inconvenient for our purpose, for the following reason. Since `lmfit`

[1] `http://www.python.org`
[2] `http://cars9.uchicago.edu/software/python/lmfit/`
[3] `http://en.wikipedia.org/wiki/Levenberg-Marquardt_algorithm`
[4] `http://en.wikipedia.org/wiki/MINPACK`

has been built for fitting a model formula to a set of data as the name suggests, it assumes that every data point has the same structure, whereas in our case, each cue has a different structure. The same can be said with other problem description programs $\texttt{des}_q$ for statistics such as the `formula` provided in `R` (R Core Team (2013)).

If the Levenberg-Marquardt algorithm for the least squares is deemed more reliable and perhaps more efficient than generic unconstrained optimization algorithms, then one should use the former whenever available. An obvious research direction, then, would be to develop a problem description software $\texttt{des}_q$ suitable for utility maximization in which each cue comes with an individual formula, rather than for fitting the same formula to a large data set.

REFERENCES

Hastie, R., Dawes, R.M., 2009. Rational Choice in an Uncertain World: The Psychology of Judgement and Decision Making, 2nd Edition. Sage Publications, Inc.

R Core Team, 2013. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
URL `http://www.R-project.org/`

Turi, D., 2001. Category Theory Lecture Notes. University of Edinburgh.
URL `http://www.dcs.ed.ac.uk/home/dt/CT/categories.pdf`

Yoneda, K., Celaschi, W., 2013. A utility function to solve approximate linear equations for decision making. Decision Making in Manufacturing and Services, **7**(1–2), pp. 5–18.
URL `http://www.dmms.agh.edu.pl/Volume_7/DMMS_2013_Yoneda_Celaschi.pdf`