Stretching Least Squares to Embed Loss Function Tables

Kiyoshi Yoneda
Faculty of Economics, Fukuoka University
Jounan-ku, Fukuoka, 814-0180 Japan
E-mail: yoneda@econ.fukuoka-u.ac.jp

Antonio Carlos Moretti
School of Applied Sciences
State University of Campinas – SP – Brazil
E-mail: antonio.moretti@fca.unicamp.br

Johan Hendrik Poker, Jr.
School of Applied Sciences
State University of Campinas – SP – Brazil
E-mail: johan.poker@fca.unicamp.br

2015-08-15

Abstract The method of least squares is extended to accommodate a class of loss functions specified in the form of function tables. The function tables are embedded into the standard quadratic loss function so that nonlinear least squares algorithms can be adopted for loss minimization. This is an alternative to a more straight-forward approach which interpolates the function tables and minimizes the resulting loss function by some generic optimization algorithm. The alternative approach has advantages over the straight-forward, such as the wider availability of the least squares programs compared to the generic optimization programs and a reduction in computational complexity. Examples are given for its application to multiplicative utility function maximization problems.

Keywords: least squares, individual behavior, inverse problems, simultaneous equa-

tions, optimization
MSC code: 90B50 Management decision making, including multiple objectives
JEL code: C44 - Operations Research; Statistical Decision Theory

1 Introduction

A number of variables $x := [\cdots\ x_i\ \cdots]$ are considered *outcomes* of any decision. The *decision* variables $\check{x} = [\cdots\ \check{x}_k\ \cdots]$ consist a part of the outcomes. For notational convenience, we let $\check{x}$ be the first part of x, so that $x_i = \check{x}_i$ for $1 \leq i \leq \dim \check{x}$:

$$x = [\cdots\ x_i\ \cdots] = [\check{x}\ \ x_{\dim \check{x}+1}\ \cdots\ x_{\dim x}]\ .$$

The *causality* relationship between the decision and the outcome variables is assumed known in the functional form

$$x = \mathrm{f}(\check{x}) = [\cdots\ \mathrm{f}_i(\check{x})\ \cdots] \tag{1}$$

where $\check{x}_i = \mathrm{f}_i(\check{x})$, and f_i is monotone with respect to each $\check{x}_k$. The solution to (1) is defined by

$$\hat{x} = \arg\min_{\check{x}} \sum_i w_i\, \ell_i(x_i) \qquad\qquad \sum_i w_i = 1 \tag{2}$$

where w_i are given *importance weights*. Depending on ℓ_i, this includes the weighted likelihood maximization and the weighted *least squares* (LS) which may be found in textbooks such as [Hansen et al.(2012)Hansen, Pereyra, and Scherer].

Since writing a mathematical expression for general ℓ_i is often difficult, we propose that the specifications be given as **subloss function tables**

$$L_i := \begin{bmatrix} x_{i\bullet} \\ y_{i\bullet} \end{bmatrix} := \begin{bmatrix} \cdots & x_{ij} & \cdots \\ \cdots & \ell_i(x_{ij}) = y_{ij} & \cdots \end{bmatrix} \qquad \begin{matrix} x_{ij} < x_{i\,j+1} \\ 0 \leq y_{ij}\ . \end{matrix} \tag{3}$$

This representation permits loss functions that are far more complex than the quadratic loss. Specifically, unlike in the least squares, the subloss function ℓ needs not be symmetric with respect to x such that $\ell(x) = 0$.

A straightforward way to process such a specification would be:

1. Interpolate the L_i to obtain ℓ_i.
2. Use a generic *unconstrained optimization* (UO) program to compute (2).

This paper develops an alternative solution method, which extends the method of LS to accommodate subloss functions given as function tables (3). The method tweaks the standard quadratic subloss function z_i^2 into the subloss function $\ell_i(x_i)$ which passes through the given knots $\{\cdots (x_{ij}, y_{i,j}) \cdots\}$ by setting a piecewise linear bijection $z_i \overset{\mathrm{x}_i}{\mapsto} x_i$:

$$\begin{array}{ccc} x_{ij} & \xrightarrow{\quad L_i \quad} & y_{ij} \\ \Big\downarrow \text{embed} & & \Big\downarrow \text{embed} \\ x_i & \overset{\ell_i}{\dashrightarrow} & y_i \\ \Big\uparrow \text{piecewise linear bijection } \mathrm{x}_i & & \Big\| \\ z_i & \xrightarrow[\text{quadratic loss}]{} & z_i^2 \end{array} \tag{4}$$

The dashed ℓ_i is uniquely determined by making the diagram commute. Figure 3 visualizes the situation in graphs.

Motivations for this approach have been:

1. LS programs are more widely available than UO programs, especially for resource-constrained computers such as controllers and signal processors.
2. LS programs tend to be more robust and faster than UO programs.
3. Engineering users are more likely to be familiar with the LS than with the UO.
4. Writing a program for the LS is usually easier than for the UO for various reasons, including the availability of test data.
5. The alternative approach is computationally straightforward one.

The method is a natural extension of the idea described in [Yoneda and Moretti(2014)], in which the function tables are limited to three entries,

$$L_i := \begin{bmatrix} x_{i1} & x_{i2} & x_{i3} \\ 1 & 0 & 1 \end{bmatrix} . \tag{5}$$

The rest of this paper is organized as follows: the bijection $z_i \overset{\mathrm{x}_i}{\mapsto} x_i$ is presented in Section 2 and discussed in Section 3. The proposed method is illustrated in Section 4 by examples of multiplicative utility maximization. Section 5 concludes the paper with remarks on deployability.

2 Extension of the least squares

The function table (3) is assumed to satisfy the following conditions:

- The table L_i has at least three entries as in (5).
- The minimum $y_{ij} = 0$ is unique; the maximum $y_{ij} = 1$ is at the both ends of the table L_i.
- The slopes connecting adjacent points are increasing,

$$\frac{y_{ij} - y_{i\,j-1}}{x_{ij} - x_{i\,j-1}} < \frac{y_{i\,j+1} - y_{ij}}{x_{i\,j+1} - x_{ij}} \qquad 1 < j < \dim x_{i\bullet} \ .$$

The proposed method is as follows:

1. Define the unique piecewise linear bijection

$$\mathbb{R}^{\dim x} \ni z \overset{\mathrm{x}}{\mapsto} x \in \mathbb{R}^{\dim x}$$

that makes (4) commute, where $\mathbb{R}$ is the set of real numbers.

2. Use a nonlinear LS software to compute

$$\hat{z} = \arg\min_{\check{z}} \sum_i w_i \, z_i^2 \qquad \text{with} \qquad \check{z} \overset{\mathrm{x}|\{\check{z}\}}{\mapsto} \check{x} \overset{\mathrm{f}}{\mapsto} x \overset{\mathrm{x}^{-1}}{\mapsto} z \ .$$

where $\mathrm{x}|\{\check{z}\}$ is x restricted to $\{\check{z}\}$.

3. Recover the solution to (2) by $\hat{z} \overset{\mathrm{x}}{\mapsto} \check{x} \overset{\mathrm{f}}{\mapsto} x$.

Thus, the LS computation proceeds as follows (where the iterations are numbered):

$$\begin{array}{ccccccc}
\check{z}^{(0)} & \xrightarrow{\mathrm{x}} & \check{x}^{(0)} & \xrightarrow{\mathrm{f}} & x^{(0)} & \xrightarrow{\mathrm{x}^{-1}} & z^{(0)} \\
 & & & & & \searrow^{\text{LS update}} & \\
\check{z}^{(1)} & \xrightarrow[\mathrm{x}]{} & \check{x}^{(1)} & \xrightarrow{\mathrm{f}} & x^{(1)} & \xrightarrow{\mathrm{x}^{-1}} & z^{(1)} \\
 & & & & & \searrow^{\text{LS update}} & \\
\check{z}^{(2)} & \xrightarrow{} & \cdots & & & &
\end{array} \tag{6}$$

To find $z \overset{\mathrm{x}}{\mapsto} x$, determine the values of z at the *knot* points x_{ij} by

$$z_{ij} := \begin{cases} -y_{ij}^{\frac{1}{2}} & j \le k \\ y_{ij}^{\frac{1}{2}} & k \le j \end{cases} \qquad \text{where} \qquad y_{ik} = 0$$

so that

$$z \overset{\mathrm{x}}{\mapsto} x = a_0 + a_1 z$$

with

$$a_0 := \frac{x_{ih} z_{i\,h+1} - x_{i\,h+1} z_{ij}}{z_{i\,h+1} - z_{ih}} \qquad\qquad a_1 := \frac{x_{i\,h+1} - x_{ih}}{z_{i\,h+1} - z_{ih}}$$

$$h := \begin{cases} 1 & z < z_{i1} \\ j & z_{ij} \le z \le z_{i\,j+1} \\ \dim x_i - 1 & z_{i\,\dim x_i \bullet} < z \end{cases}$$

For an example, see the lower left graph in Figure 3.

3 Discussion

3.1 Computational complexity

Since the causality $\check{x} \overset{\mathrm{f}}{\mapsto} x$ is described in terms of x rather than z, the bijection $z \overset{\mathrm{x}}{\mapsto} x$ and its inverse $x \overset{\mathrm{x}^{-1}}{\mapsto} z$ have to be invoked for each iteration in the LS algorithm, as illustrated in (6).

The piecewise linear bijection $z_i \overset{\mathrm{x}_i}{\mapsto} x_i$ induces a piecewise quadratic interpolation $x_i \overset{\ell}{\mapsto} y_i$ of L_i , as illustrated in Figure 2. The straightforward way mentioned in Section 1 can be used combining this piecewise quadratic interpolation with UO:

$$\begin{array}{ccccccc} \check{x}^{(0)} & \xrightarrow{\mathrm{f}} & x^{(0)} & \xrightarrow{\text{quadratic interpolation}} & y^{(0)} & & \\ & & & \text{UO update} \swarrow & & & \\ \check{x}^{(1)} & \underset{\mathrm{f}}{\longleftarrow\!\!\shortrightarrow} & x^{(1)} & \xrightarrow{\text{quadratic interpolation}} & y^{(1)} & & \\ & & & \text{UO update} \swarrow & & & \\ \check{x}^{(2)} & \longleftarrow\!\!\shortrightarrow & \cdots & & & & \end{array} \tag{7}$$

It is fair to compare the computational complexities of (6) against (7).

For each iteration in (6), every decision variable invokes one x which involves one multiplication, since the conversion is of the form $\check{x}_k = a_0 + a_1 \check{z}_k$. This amounts to $\dim \check{x}$ multiplications. Likewise, each variable x_i invokes one multiplication to compute x^{-1} for the form $\mathrm{z}_i = (x_i - a_0) a_1^{-1}$. This amounts to $\dim x$ multiplications. Hence, the total is $\dim \check{x} + \dim x$ multiplications per iteration. On the other hand, for each iteration in (7), every variable invokes one quadratic interpolation which involves four multiplications, since the computation is of the form $y_i = b_0 + b_1 x_i + b_2 x_i^2$. Hence, the total is $4 \dim x$ multiplications per iteration.

Thus, a single LS iteration involves fewer multiplications than a single UO iteration.

3.2 Differentiability

At this point, the result is that even if UO updated equally fast as LS and if UO converged equally fast as LS, (6) still holds an advantage over (7) in terms of computational complexity. However, this cannot be taken as the final conclusion.

The piecewise quadratic interpolation ℓ_i is smooth except at the *knot* points (x_{ij}, y_{ij}) where the derivatives do not exist. The exception is at the minimum $\ell_k(x_k) = 0$, in which case $\left.\frac{d\ell_i}{dx_i}\right|_{x_i=x_{ik}} = 0$ but not generally twice differentiable. If this problem is to be avoided, either a more sophisticated interpolation of L_i with UO or a more sophisticated bijection $z \overset{\mathrm{x}}{\mapsto} x$ with LS would have to be used, which wold take care of differentiability at the knots. An obvious candidate would be splines, which would require at least a cubic polynomial adding complexity to the computation.

This concern is not addressed herein but passed over to the LS program adopted. The problem has yet to surface in practice perhaps because our experience is limited. The Levenburg-Marquardt algorithm [Gavin(2013)], adopted in many of the LS programs, tends to be robust in practice. Even in case the available LS programs are not robust enough and other algorithms have to be considered, it is easier to code a special-purpose LS program than a general-purpose UO program.

4 Examples

Simple examples are included herein. All examples are about *multiplicative utility maximization* rather than additive because of a higher demand. The current implementation is in R [R Core Team(2014)] with `nlmrt` library [Nash(2012)] which was preferred over the standard `nls` nonlinear least squares library because of its robustness.

4.1 Peanuts and beer

This is a slight modification of the example used in [Yoneda and Celaschi(2013), Yoneda and Moretti(2014)] in which you wish to enjoy eating and drinking without spending too much and preventing the risk of getting too fat.

You wish to eat peanuts drinking beer minding cost and gaining weight. Ideally, you want to eat 15 g peanuts with 350 ml beer for 2 currency units gaining 50 kcal. But it's acceptable if peanuts is between 5 and 15 g, beer is between 200 and

500 ml, cost is between 0 and 10 currency units, and energy is between 0 and 200 kcal. As for the beer quantity you have a more detailed specification as below:

peanuts in [g = grams], importance weight $w = 0.2$

Amount x [g]	5	15	20
Subutility u	1	5	1

beer in [ml = milliliters], $w = 0.3$

x [ml]	200	300	350	450	500
u	1	9	10	8	1

cost in [cu = currency units], $w = 0.1$

x [cu]	0	2	10
u	1	5	1

energy in [kcal = kilocalories], $w = 0.4$

x [kcal]	0	50	200
u	1	10	1

The utility function table for `beer` is plotted in 1. The dashed lines connecting the knots are for the linear interpolation, which is not used hereafter. The cost's utility here ought to be monotone decreasing, so the first part of x is to be considered a dummy. The cost of 2 [cu] is considered twice as desirable as 10 [cu]. Similar interpretation can be made for the energy.

We know the causality to be

$$\begin{bmatrix} x_{\text{peanuts}} \\ x_{\text{beer}} \\ x_{\text{cost}} \\ x_{\text{energy}} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & \\ & 1 \\ 1/50 & 1/100 \\ 592/100 & 142/350 \end{bmatrix} \begin{bmatrix} x_{\text{peanuts}} \\ x_{\text{beer}} \end{bmatrix} .$$

The major differences in treatment from the previous papers are:

- The subloss function table may now have a description longer than three; for instance, beer now has five points specified.
- The utility function is now multiplicative rather than additive, *viz.* there is no fun in eating only peanuts without beer or drinking beer without peanuts; they work together.

The transformation from the multiplicative utility to the additive loss is done by

$$y_{ij} := \frac{\log \frac{u_{ij}}{\max u_{i\bullet}}}{\log \frac{\min u_{i\bullet}}{\max u_{i\bullet}}} \qquad\qquad u_{i\bullet} := [\cdots\ u_{ij}\ \cdots]$$

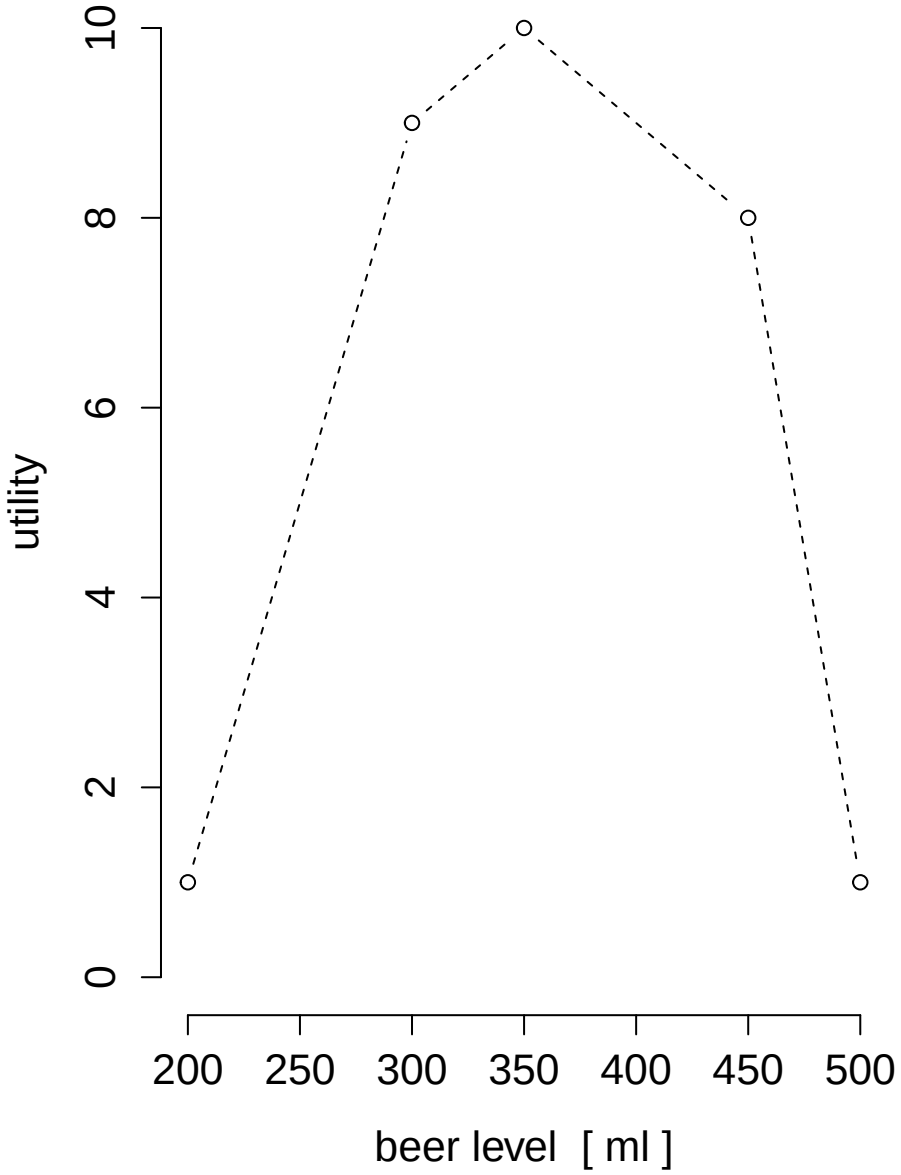

Figure 1: Utility specification $x_{\text{beer } j} \mapsto u_{\text{beer } j}$

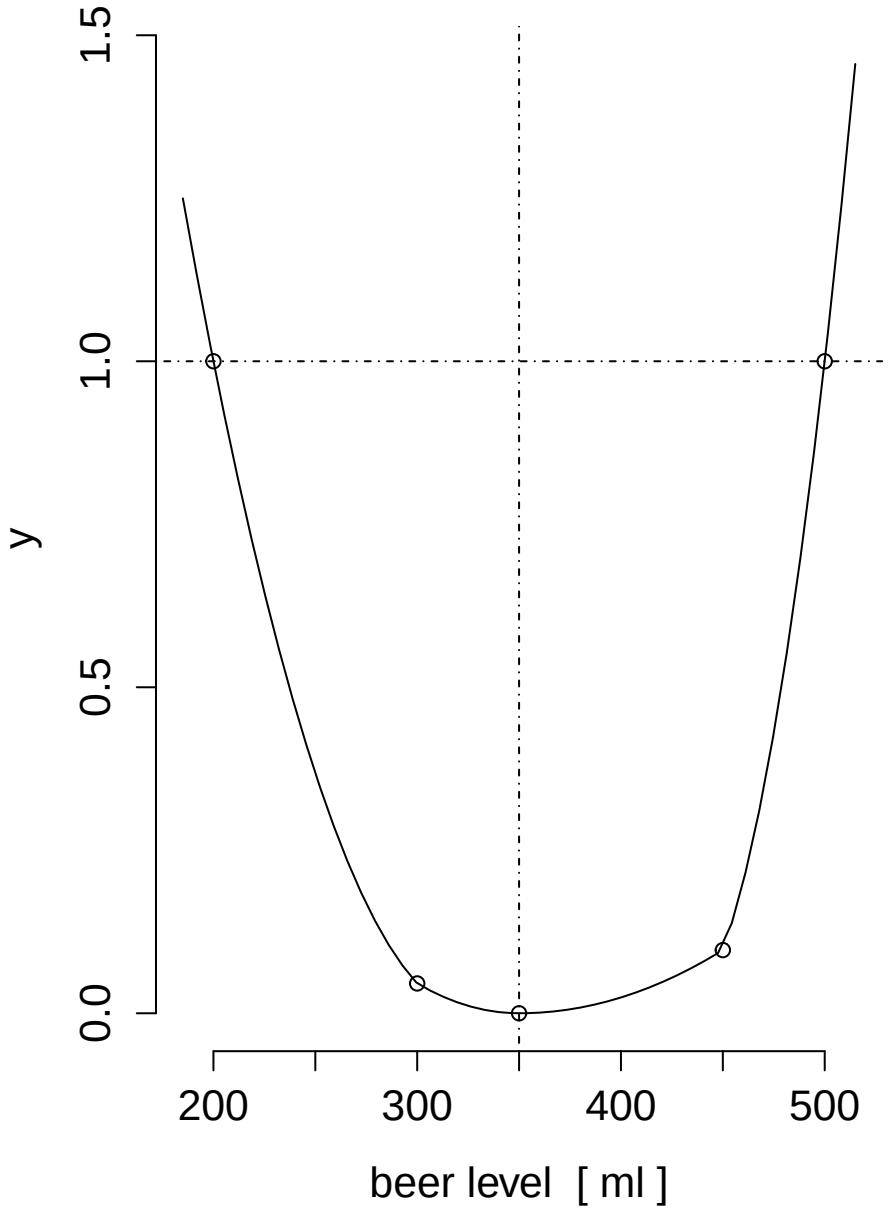

Figure 2: Subloss $x \overset{\ell}{\mapsto} y$ for `beer` with knots

so that $0 \le y_{ij} \le 1$. For `beer`, this transforms Figure 1 to Figure 2.

The bijection $z \overset{\mathrm{x}}{\mapsto} x$ is illustrated in Figure 3 for the case of $i =$ `beer` . In the rest of this paragraph, subscripts i are dropped for clarity. The upper left graph shows the standard subloss function z^2 to be minimized by the LS. The upper right graph is the same as Figure 2 showing the subloss function $x \overset{\ell}{\mapsto} y$ for `beer` . The lower left graph is for the piecewise linear $z \overset{\mathrm{x}}{\mapsto} x$. The lower right graph is just the identity transformation $x \overset{1}{\mapsto} x$ to compose $z \overset{\mathrm{x}}{\mapsto} x$ and $x \overset{\ell}{\mapsto} y$. The dotted lines show the correspondence among the values at the knots.

The *residual* to be minimized by the LS is

$$r := \begin{bmatrix} w_{\text{peanuts}} & & & \\ & w_{\text{beer}} & & \\ & & w_{\text{cost}} & \\ & & & w_{\text{energy}} \end{bmatrix}^{\frac{1}{2}} \begin{bmatrix} z_{\text{peanuts}} \\ z_{\text{beer}} \\ \mathrm{x}^{-1}(x_{\text{cost}}) \\ \mathrm{x}^{-1}(x_{\text{energy}}) \end{bmatrix} .$$

If one eats and drinks as she wishes, she gains energy beyond the permissible range:

$$\begin{bmatrix} 0 \\ 0 \\ 2 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & \\ & 1 \\ \frac{1}{50} & \frac{1}{100} \\ \frac{592}{100} & \frac{142}{350} \end{bmatrix} \begin{bmatrix} x_{\text{peanuts}} = 15 \text{ [g]} \\ x_{\text{beer}} = 350 \text{ [m]} \end{bmatrix} = \begin{bmatrix} x_{\text{peanuts}} = 15 \text{ [g]} \\ x_{\text{beer}} = 350 \text{ [m]} \\ x_{\text{cost}} = 5.8 \text{ [cu]} \\ \boxed{x_{\text{energy}} = 231 \text{ [kcal]}} \end{bmatrix} .$$

Nonlinear LS or generic UO programs yield the best solution

Item	x	[unit]	z
peanuts	9	[g]	-0.614
beer	280	[ml]	-0.374
cost	5	[cu]	0.373
energy	166	[kal]	0.773

Thus, the recommendation is to reject some peanuts and be modest on beer to keep the energy in the permissible range.

4.2 Fuzzy guess

We heard through the grapevine that L. A. Zadeh, the founder of the fuzzy theory, mentioned the following problem in one of his talks.

> A box contains about 20 balls of various sizes. Most of them are large. The number of large balls is several times larger than the number of small balls. How many small balls are there in the box?

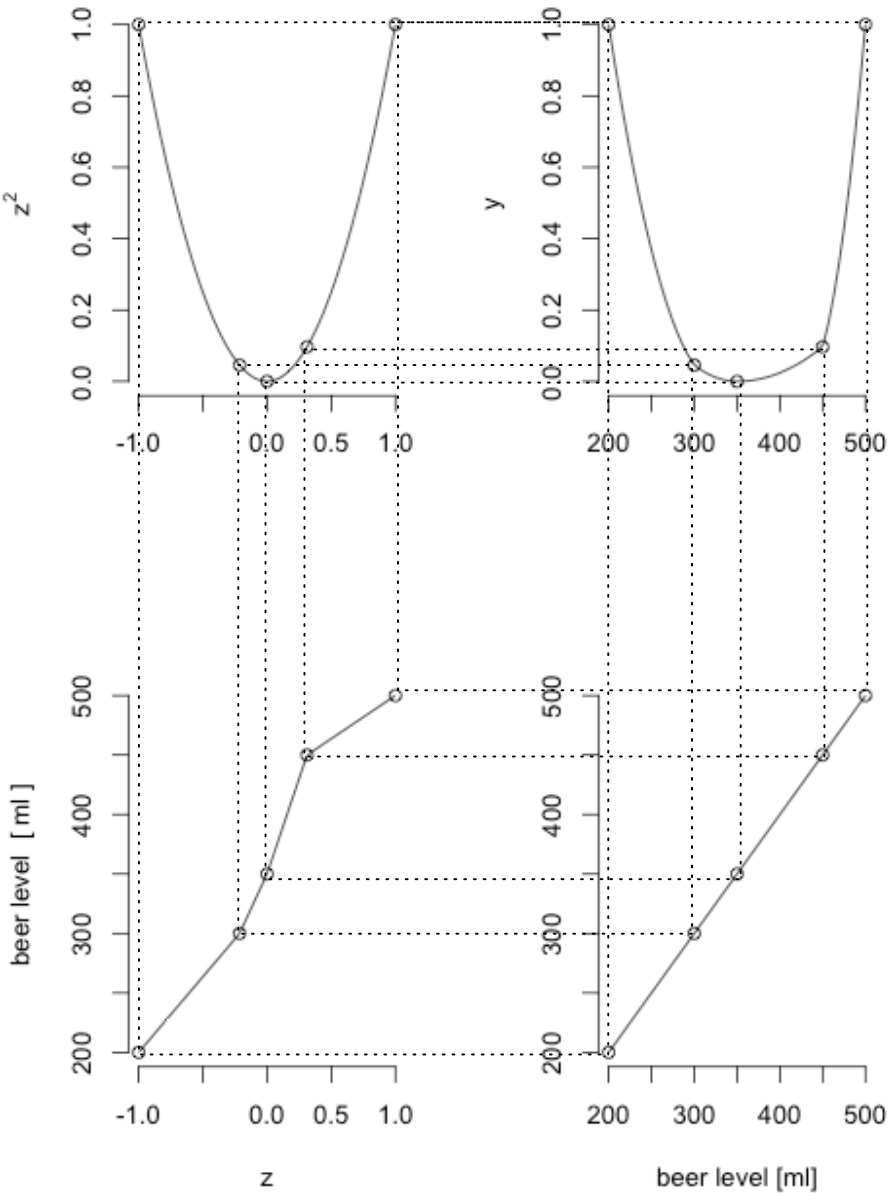

Figure 3: The piecewise linear bijection $z \overset{x}{\mapsto} x$ for `beer`

A solution may be found by feeding enough prior knowledge.

The utility specifications are:

small in [balls], $w = 0.1$

x [balls]	0	1	10	19	20
u	1	3	4	3	1

several in [none], $w = 0.2$

x [none]	3	5	8
u	1	2	1

all in [balls], $w = 0.7$

x [balls]	15	17	19	20	70
u	1	5	9	10	1

If there are 10 small balls and "several" means 5, then the total number of balls would be 50, a lot more than "about 20." A multiplicative utility is assumed. The causality is

$$\begin{bmatrix} x_{\text{small}} \\ x_{\text{several}} \\ x_{\text{all}} \end{bmatrix} = \begin{bmatrix} x_{\text{small}} \\ x_{\text{several}} \\ (1 + x_{\text{several}})\ x_{\text{small}} \end{bmatrix} .$$

About the two decision variables, we are not confident; about the pure outcome, we are relatively confident.

The residual is

$$r := \begin{bmatrix} w_{\text{small}} & & \\ & w_{\text{several}} & \\ & & w_{\text{all}} \end{bmatrix}^{\frac{1}{2}} \begin{bmatrix} z_{\text{small}} \\ z_{\text{several}} \\ \mathrm{x}^{-1}(x_{\text{all}}) \end{bmatrix} .$$

The solution is

	Item	x	[unit]	z
$\rightarrow$	small	3.5	[balls]	-0.328
	several	5.0	[none]	-0.010
	all	21.0	[balls]	0.020

So our guess is that there are 3 or 4 small balls.

The guess could be done by the *weighted log-likelihood maximization* as well, resulting in a similar numerical computation but in a quite different interpretation involving probability.

4.3 Plastics production

This problem is the same as in [Yoneda and Celaschi(2013)] except that the utility function is now multiplicative, the marginal utilities are more detailed, and the importance weights are set to all equal.

> You run a factory that produces plastics, hard and soft. You need to decide how much of each kind you will produce this coming week.
>
> The sales department says that they want 4 and 6 [t = tons] of hard and soft plastics. Since you have product stocks and some space in the warehouse, the quantities produced would be acceptable if they are between 3.5 and 5 [t] for hard and 4 and 7 [t] for soft plastics.
>
> The purchase department says that they will have 10 [t] raw material mix available, which serves to produce both hard and soft plastics. Considering the relationship with the supplier, it is undesirable to order less than 8 [t]. Also, it will be difficult to prepare more than 13 [t] even considering extra purchases not only from the supplier but also from the market.
>
> The personnel department says that they have 8 [p = people] available for the next week. It is possible, however, to adjust it between 7 and 9 [p] by reducing or extending work hours without hiring or firing.
>
> The engineering department says that they have 15 [m = machines] leased for production but can reduce to 12 [m] or increase to 17 [m] by adjusting operation time and machine speed.
>
> The production department says that, in order to produce 1 [t] of hard plastics, you need 2 [t] material, 1 [p] labor, and 3 [m] machines, while to produce 1 [t] of soft plastics, you need 1 [t] material, 1 [p] labor, and 1 [m] machine.

The requirements are

hard in [t = tons], $w = 0.2$

x [t]	3.5	4	5
u	1	3	1

soft in [t], $w = 0.2$

x [t]	4	6	7
u	1	3	1

mix in [t], $w = 0.2$

x [t]	8	10	13
u	1	5	1

labor in [p = persons], $w = 0.2$

x [p]	7	8	9
u	1	10	1

machines in [m = machines], $w = 0.2$

x [m]	12	15	17
u	1	4	1

The causality is

$$\begin{bmatrix} x_{\text{hard}} \\ x_{\text{soft}} \\ x_{\text{mix}} \\ x_{\text{labor}} \\ x_{\text{machines}} \end{bmatrix} = \begin{bmatrix} 1 & \\ & 1 \\ 2 & 1 \\ 1 & 1 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} x_{\text{hard}} \\ x_{\text{soft}} \end{bmatrix} .$$

If the production is done in the way that the sales considers optimal, then mix, labor, and machines will all be short by one unit. The best compromise is to reduce a little of the production of hard plastics and more of soft plastics to keep the resources needed within the permissible ranges:

Item	x	[unit]	z
hard	3.8	[t]	-0.465
soft	4.3	[t]	-0.849
mix	11.8	[t]	0.612
labor	8.1	[p]	0.069
machines	15.6	[m]	0.302

5 Concluding remarks

This paper extended the method of least squares to accommodate the loss specification given as function tables.

We hope to apply the method to large-scale real-world problems to assess its practicability. The bottleneck to deploy this method for those not versed in mathematics is not the description of subloss tables (3) but remains to be the description of causality relationships (1).

References

[Gavin(2013)] Gavin, H. P., 2013. The levenberg-marquardt method for nonlinear least squares curve-fitting problems.
URL `http://people.duke.edu/~hpgavin/ce281/lm.pdf`

[Hansen et al.(2012)Hansen, Pereyra, and Scherer] Hansen, P. C., Pereyra, V., Scherer, G., 2012. Least Squares Data Fitting with Applications. Johns Hopkins University Press.

[Nash(2012)] Nash, J. C., 2012. nlmrt-vignette. R Foundation for Statistical Computing.

[R Core Team(2014)] R Core Team, 2014. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
URL http://www.R-project.org/

[Yoneda and Celaschi(2013)] Yoneda, K., Celaschi, W., 2013. A utility function to solve approximate linear equations for decision making. Decision Making in Manufacturing and Services 7, 3–16.
URL http://www.dmms.agh.edu.pl/Volume_7/DMMS_2013_Yoneda_Celaschi.pdf

[Yoneda and Moretti(2014)] Yoneda, K., Moretti, A. C., 2014. Maximization of an asymmetric utility function by the least squares. Decision Making in Manufacturing and Services 8 (1–2).
URL https://journals.agh.edu.pl/dmms/article/view/738