G A RAS-like method for balancing data

Consider the 2-dimensional data as outlined in table 1 where $v_{i,j}$ indicate cell values and R_i/ C_j indicate row/column sums.

Table 1: A representation of 2D data

	x	**y**	
a	$v_{a,x}$	$v_{a,y}$	R_a
b	$v_{b,x}$	$v_{b,y}$	R_b
c	$v_{c,x}$	$v_{c,y}$	R_c
	C_x	C_y	

We consider the case where some values in this table is fixed at zero for some reason (negative domains/small values). The following procedure attempts to cells in a way that keeps row/column sums intact and leaves the new data as true to the original as possible.

1. Fix data

We start by defining a couple of auxiliary variables:

- Let z^0 denote initial data values in general.
- Let $\overline{z}$ denote manual data adjustments to the data – i.e. fixing a value to zero $\overline{v}_{a,x} = 0$, and $\mathcal{D}$ denote the subset of (i,j) that are fixed.
- Given these adjustments, define the new step with a superscript 1:

$$v_{i,j}^1 = \begin{cases} \overline{v}_{i,j}, & \text{if } (i,j) \in \mathcal{D} \\ v_{i,j}^0, & \text{else.} \end{cases}$$

- Define the column, row distributions $(\gamma_{i,j}, \omega_{i,j})$ as the share of the respective value:

$$\gamma_{i,j} = \frac{v_{i,j}^1}{\tilde{C}_j^1}, \qquad \tilde{C}_j^1 \equiv C_j^1 - \sum_{i' \in \mathcal{D}_j} \overline{v}_{i',j},$$
$$\omega_{i,j} = \frac{v_{i,j}^1}{\tilde{R}_i^1}, \qquad \tilde{R}_i^1 \equiv R_i^1 - \sum_{j' \in \mathcal{D}_i} \overline{v}_{i,j'}.$$

  This ensures that summing $(\gamma_{i,j}, \omega_{i,j})$ over rows/columns that are not fixed by data $(i,j) \notin \mathcal{D}$ sums to 1.

- Define the percentage change in row/column sums from the manual data adjustments:

$$\Delta R_i \equiv R_i^1 - R_i^0, \qquad \Delta r_i \equiv \frac{\Delta R_i}{\tilde{R}_i^1}$$
$$\Delta C_j \equiv C_j^1 - C_j^0, \qquad \Delta c_j \equiv \frac{\Delta C_j}{\tilde{C}_j^1}$$

2. Minimize distortions to data

To fix ideas, let's start by considering the case where we only had to achieve the same column sum as before the change. In this case, we would suggest using the following adjustment procedure for all $(i,j) \notin \mathcal{D}$:

$$\begin{aligned} v_{i,j} &= v^1_{i,j} - \gamma_{i,j} \Delta C_j \\ &= v^1_{i,j} \left(1 - \Delta c_j\right). \end{aligned}$$

The idea is that the row changes are mapped using the distributions $\gamma_{i,j}, \omega_{i,j}$ as weights. Consider the case in table 2 and assume that we want to fix $v_{a,x} = 0$. If we only cared about keeping the column sums constant, our approach would suggest that we double both $v_{b,x}$ and $v_{c,x}$ – i.e. increasing the remaining values proportionally. As the tables above illustrate, however, this approach does not keep row sums intact. As a natural extension – if it was feasible – we would ideally extend the formula above to include row-sums:

Table 2: A representation of 2D data – Example 1

Step 0 – Initial data

	x	y	
a	1	2	3
b	1/3	2/3	1
c	2/3	4/3	2
	2	4	

⇒

Step 1 – Manual adj.

	x	y	
a	-	2	2
b	1/3	2/3	1
c	2/3	4/3	2
	1	4	

⇒

Step 2 – Column sol.

	x	y	
a	-	2	2
b	2/3	2/3	4/3
c	4/3	4/3	8/3
	2	4	

$$v_{i,j} = v^1_{i,j} \left(1 - \Delta c_j - \Delta r_i\right).$$

This, however, does not (generally at least) fix the issue of keeping row/column sums fixed. Instead, we consider the more flexible quadratic program:

$$\min_{\{\eta^r_{i,j}, \eta^c_{i,j}, v_{i,j}\}_{(i,j)\notin\mathcal{D}}} \sum_{(i,j)\notin\mathcal{D}} \left[\left(\eta^r_{i,j} - 1\right)^2 + \left(\eta^c_{i,j} - 1\right)^2\right] \tag{14a}$$

$$\text{s.t. } v_{i,j} = v^0_{i,j}\left(1 - \eta^r_{i,j}\Delta r_i - \eta^c_{i,j}\Delta c_j\right), \qquad \forall (i,j) \notin \mathcal{D} \tag{14b}$$

$$C^0_j = \sum_{i\in\mathcal{D}_j} \overline{v}_{i,j} + \sum_{i\notin\mathcal{D}_j} v_{i,j}, \qquad \forall j \tag{14c}$$

$$R^0_i = \sum_{j\in\mathcal{D}_i} \overline{v}_{i,j} + \sum_{j\notin\mathcal{D}_i} v_{i,j}, \qquad \forall i \tag{14d}$$

$$v_{i,j} \geq 0, \qquad \forall (i,j) \notin \mathcal{D}, \tag{14e}$$

where the last inequality can be dropped if this is not essential for the type of data. Naturally, there are still bounds to how many manual adjustments we can impose on the data, but this can be solved relatively flexibly. In particular, note that – if this is feasible – this simply chooses the optimum $\eta^r_{i,j} = \eta^c_{i,j} = 1$.

3. Ensuring the problem is feasible

Let n denote the number of active values i.e. $\#(i,j) \notin \mathcal{D}$, n_c the number of column constraints, and n_r the number of row constraints. The quadratic problem can be reduced to identifying $2n$ variables $(\eta^r_{i,j}, \eta^c_{i,j})$ such that the rows/column sums hold. Thus, feasibility consists of $n_r + n_c$ linear constraints in $2n$ variables. Thus, for feasibility, we need at least one unique element $v_{i,j}$ in the active set for each i and for each j that is constrained.

Let us assume that the objective of this adjustment is to obtain a sparse, non-negative matrix. One way to identify a feasible active set is then:

i. Let $\boldsymbol{v}^0$ denote the initial data matrix. Identify the maximum of $v_{i,j}$ for each i - $\boldsymbol{v}^i_{max}$. Define $\tilde{\boldsymbol{v}}^0 = \boldsymbol{v}^0 \backslash \boldsymbol{v}^i_{max}$.

ii. Identify the maximum of $v_{i,j}$ for each j from $\tilde{v}$. Define $\tilde{\boldsymbol{v}}^1 = \tilde{v}^0 \backslash \boldsymbol{v}^j_{max}$.

iii. Given that $(\boldsymbol{v}^i_{max}, \boldsymbol{v}^j_{max})$ are in the active set, we are certain of feasibility.

In this crude algorithm the order of row/columns, unfortunately, may make a difference in the final active set; this effect is, however, minor, as long as we do no not remove large values.