Phonology **29** (2012). Supplementary materials

Convergence of error-driven ranking algorithms

Giorgio Magri
CNRS, Université Paris 7

Supplementary materials

Appendix A: Extension to arbitrary initial ranking vectors

Throughout the paper, I have investigated convergence in the case where the initial RVs θ_1^{init}, …, θ_n^{init} were all identical, say all equal to zero (the actual value does not really matter). In this appendix, I show how to obtain error bounds for demotion-only and calibrated re-ranking rules in the case of an arbitrary initial ranking vector $\boldsymbol{\theta}^{\text{init}} = (\theta_1^{\text{init}}, \ldots, \theta_n^{\text{init}})$. It turns out that the properties of the initial ranking vector that are relevant for the error-bounds can be extracted through the quantity $\Delta(\boldsymbol{\theta}^{\text{init}})$ defined in (71), namely the sum of the difference between each initial RV θ_k^{init} and the smallest RV $\min_{h=1} \theta_h^{\text{init}}$. Intuitively, this quantity measures how scattered the initial RVs are. In fact, $\Delta(\boldsymbol{\theta}^{\text{init}})$ is null for the case of identical initial RVs, small for RVs close to each other and large if there are some RVs that are very small and some other RVs that are very large.

(71) $$\Delta(\boldsymbol{\theta}^{\text{init}}) = \sum_{k=1}^{n} (\theta_k^{\text{init}} - \min_{h=1} \theta_h^{\text{init}})$$

1 The crucial invariant

Tesar & Smolensky's (generalised) Fact 2 from §6.2 provides an invariant for the current RVs entertained by the EDRA in the case of null initial RVs. It says that the current RVs can never become much smaller than zero, as long as the EDRA only demotes the loser-preferring constraints that need to be demoted, i.e. the undominated ones. The reasoning trivially extends to arbitrary initial ranking values, yielding the following further generalisation of Fact 2: the current ranking values can never become much smaller than the smallest initial ranking value.

Fact 2 (*further generalised*)

Assume that the input ERC matrix is consistent with a ranking. Without loss of generality, assume that this ranking is $\mathrm{C}_1 \gg \mathrm{C}_1 \gg \ldots \gg \mathrm{C}_n$. Let $\theta_1, \ldots, \theta_n$ be the current RVs entertained by an EDRA in a generic run on those input ERCs, up to a generic time, starting from arbitrary initial RVs $\theta_1^{\mathrm{init}}, \ldots, \theta_n^{\mathrm{init}}$. Assume that the re-ranking rule used by the EDRA demotes by 1 only the currently undominated loser-preferrers (no assumptions are made on constraint promotion). The current RVs thus satisfy condition (72) for every $k = 1, \ldots, n$. That is, the RV θ_k of the constraint C_k assigned to the kth stratum (with the 1st stratum being the top one) never drops by more than $(k-1)$ below the smallest initial RV $\min_{h=1,\ldots,n} \theta_h^{\mathrm{init}}$.

(72) $\theta_k \geq \min\limits_{h=1,\ldots,n} \theta_h^{\mathrm{init}} - (k-1)$

2 Demotion-only re-ranking rules

Consider the demotion-only re-ranking rule (17) from §3, repeated in (73).

(73) a. Decrease the RV of each undominated loser-preferrer by 1.
b. Do nothing to the current RV of the other constraints.

At least one constraint is demoted at each update. Hence, the total number T of updates is at most the sum of the number of times C_1 has been demoted, and the number of times C_2 has been demoted, etc, as stated in (74a). Each time constraint C_k is demoted, it is demoted by 1, and it is never promoted. Hence the number of times that constraint C_k has been demoted up to the time considered is equal to the distance $\theta_k^{\mathrm{init}} - \theta_k$ between its initial RV θ_k^{init} and its current RV θ_k, as stated in (74b). The inequality (72) says that θ_k sits between θ_k^{init} and $\min_{h=1,\ldots,n} \theta_h^{\mathrm{init}} - (k-1)$, as depicted in (75). Thus the distance between the latter two points upper bounds the distance $\theta_k^{\mathrm{init}} - \theta_k$, as stated in (74c). Finally, step (e) follows from definition (71) of the constant $\Delta(\boldsymbol{\theta}^{\mathrm{init}})$ and from the identity $\sum_{k=1}^{n}(k-1) = n(n-1)/2$.

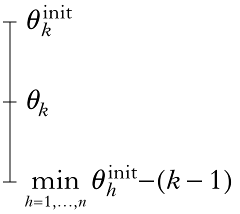
(74) a. $T \leq \sum\limits_{k=1}^{n} (\text{number of demotions of } \mathrm{C}_k)$

b. $= \sum\limits_{k=1}^{n} (\theta_k^{\mathrm{init}} - \theta_k)$

c. $\leq \sum\limits_{k=1}^{n} (\theta_k^{\mathrm{init}} - (\min\limits_{h=1,\ldots,n} \theta_h^{\mathrm{init}} - (k-1)))$

d. $= \sum\limits_{k=1}^{n} (\theta_k^{\mathrm{init}} - \min\limits_{h=1,\ldots,n} \theta_h^{\mathrm{init}}) + \sum\limits_{k=1}^{n} (k-1)$

e. $= \Delta(\boldsymbol{\theta}^{\mathrm{init}}) + \frac{1}{2} n(n-1)$

(75) 

We have thus proven the following extension of Theorem 1 from null to arbitrary initial RVs. Recall that, if the initial RVs are all identical (say, all null), then $\Delta(\boldsymbol{\theta}^{\text{init}}) = 0$. In this case, the bound $\Delta(\boldsymbol{\theta}^{\text{init}}) + n(n-1)/2$ provided by the following theorem thus reduces to the bound $n(n-1)/2$ already obtained in §3.

> Theorem 1 (extended to arbitrary initial ranking vectors)
> The EDRA (Fig. 5) with the demotion-only re-ranking rule (73) run on a consistent input ERC matrix corresponding to n constraints starting from an arbitrary initial ranking vector $\boldsymbol{\theta}^{\text{init}} = (\theta_1^{\text{init}}, \ldots, \theta_n^{\text{init}})$ can perform at most $\Delta(\boldsymbol{\theta}^{\text{init}}) + n(n-1)/2$ errors before converging.

The bound $\Delta(\boldsymbol{\theta}^{\text{init}}) + n(n-1)/2$ on the worst-case number of errors is tight, as shown by the same example in (26) in the paper with the ERCs fed in the fixed order $\mathbf{a}_1 \rightarrow \mathbf{a}_2 \rightarrow \mathbf{a}_3$ and with the initial ranking vector $\boldsymbol{\theta}^{\text{init}} = (4, 3, 2, 1)$.

3 Calibrated re-ranking rules

Consider next the calibrated demotion/promotion re-ranking rule (51) from §6, repeated in (76).

(76) a. Decrease the RV of each of the l undominated loser-preferrers by 1.
b. Increase the RV of each of the w winner-preferrers by $p = l/(w+1)$.

The invariant (72) ensures that the sum of the current RVs can be lower bounded, as in (77).

(77) $$\sum_{k=1}^{n} \theta_k \geq \sum_{k=1}^{n} \Big(\min_{h=1,\ldots,n} \theta_h^{\text{init}} - (k-1) \Big) = n \min_{h=1,\ldots,n} \theta_h^{\text{init}} - \tfrac{1}{2} n(n-1)$$

As seen in §6.3, the sum of the current RVs is decreased by at least $1/n$ with every update. After T updates, it has thus decreased by at least T/n from the sum of the initial RVs, as stated in (78).

(78) $$\sum_{k=1}^{n} \theta_k \leq \sum_{k=1}^{n} \theta_k^{\text{init}} - \frac{T}{n}$$

Combining the two inequalities (77) and (78), I conclude that the number T of updates must be smaller than $n\Delta(\boldsymbol{\theta}^{\mathrm{init}}) + n^2(n-1)/2$. We have thus proven the following extension of Theorem 2 from null to arbitrary initial RVs.

> Theorem 2 (extended to arbitrary initial ranking vectors)
> The EDRA (Fig. 5) with the calibrated promotion/demotion re-ranking rule (76) run on a consistent input ERC matrix corresponding to n constraints starting from an arbitrary initial ranking vector $\boldsymbol{\theta}^{\mathrm{init}} = (\theta_1^{\mathrm{init}}, \ldots, \theta_n^{\mathrm{init}})$ can perform at most $n\Delta(\boldsymbol{\theta}^{\mathrm{init}}) + n^2(n-1)/2$ errors before converging.

In the case of an arbitrary initial ranking vector as well, the error-bound for the calibrated case is worse by a factor of n than the error-bound for the demotion-only case.

Appendix B: Convergence of a generic calibrated re-ranking rule

In §6, I looked for concreteness at a specific calibrated re-ranking rule, namely the one in (51) (= (76)), which demotes each of the l undominated loser-preferrers by 1 and promotes each of the w winner-preferrers by $l/(w+1)$. Here I look at the generic calibrated re-ranking rule in (79).

(79) a. Decrease the RV of each of the l loser-preferrers by 1.
b. Increase the RV of each of the w winner-preferrers by $p = l/(w+\delta)$.

This re-ranking rule is calibrated as long as $\delta > 0$. Indeed, the distance of the promotion amount p from the calibration threshold l/w is controlled by the constant δ: the larger δ, the smaller the promotion amount p is with respect to the calibration threshold. In particular, the case $\delta = 1$ corresponds to re-ranking rule (51), and the case where δ goes to infinity corresponds to the demotion-only case $p = 0$ considered in §3.

The reasoning for the case $\delta = 1$ presented in §6 trivially extends to an arbitrary $\delta > 0$, yielding the following generalisation of Theorem 2 of §6.

> Theorem 2 (extended to arbitrary calibrated promotion amounts)
> An EDRA with the general calibrated re-ranking rule (79) run on a consistent input ERC matrix corresponding to n constraints starting from null initial RVs can perform at most the number of mistakes in (80) before converging, where W is the largest number of winner-preferrers over all input ERCs.

(80) $\frac{1}{2}\frac{W+\delta}{\delta}n(n-1)$

Proof. With every update, the sum of the current RVs is decreased by l. It is furthermore increased by $wl/(w+\delta)$, as each of the w winner-preferrers is promoted by $l/(w+\delta)$. In the end, the sum of the current RVs is thus decreased by $l-(wl/(w+\delta)) = \delta l/(w+\delta)$. As the number l of undominated loser-preferrers is at least 1 and the number w of winner-preferrers is at most W, I conclude that the sum of the current RVs is decreased by at least $T(\delta/(W+\delta))$ after T updates. On the other hand, the sum of the current RVs starts at zero and can never become smaller than $-n(n-1)/2$, by the generalised Fact 2 stated in §6.2. In conclusion, the number of updates T in the case of the re-ranking rule (79) must satisfy the inequality $T(\delta/(W+\delta)) \leq n(n-1)/2$, which yields the error-bound in (80). □

As there is a total of n constraints, and each ERC must have at least a loser-preferrer (ERCs that have no loser-preferrers cannot ever trigger any update and can therefore be ignored), then the largest number W of winner-preferrers is upper bound by $n-1$, and the bound (80) becomes (81).

(81) $\frac{1}{2}\frac{n-1+\delta}{\delta}n(n-1)$

The bound (81) for $\delta=1$ gives back the bound $n^2(n-1)/2$ of the original Theorem 2 in §6. As δ increases and the promotion amount $p=l/(w+\delta)$ thus gets smaller relative to the calibration threshold l/w, the bound (81) on the number of mistakes decreases, ensuring faster convergence. In the limit of δ going to infinity, the coefficient $(\mathrm{n}-1+\delta)/\delta$ goes to 1, and the bound (81) thus becomes the bound $n(n-1)/2$ already obtained in Theorem 1 of §3 for the case with null $p=0$ promotion amount.

Note that the cubic rather than quadratic growth in n of the bound (81) comes from the fact that I have upper bounded the largest number W of winner-preferrers in a generic input ERC, with $n-1$. But in most applications, W is much smaller than $n-1$, as the winner and loser forms that correspond to an ERC differ only in a few respects, and thus most of the constraints are even. Furthermore, if the loser forms are properly chosen so that the input ERCs have as few winner-preferrers as possible, then W might be forced into a constant in certain applications. In that case, the error-bound (80) for calibrated promotion grows only quadratically in the number of constraints n, just like the bound $n(n-1)/2$ for the demotion-only case.

Appendix C: Why EDRAs cannot loop

Consider the general re-ranking rule in (82).

(82) a. Decrease the RV of the loser-preferrer by 1.
b. Increase the RV of each winner-preferrer by $p>0$.

Throughout this section, I assume that the promotion amount p is never null. I show that the EDRA with the re-ranking rule (82) cannot loop on consistent input ERC matrices, as stated in Fact 5, repeated below. The proof is based on a connection between OT-consistency and CONIC INDEPENDENCE.

> FACT 5
>
> If the input ERC matrix is consistent, the EDRA (Fig. 5) with any promotion/demotion re-ranking rule of the form (82) can never loop back to a current ranking vector that it had previously made a mistake on in that same run.

Let m be the total number of input ERCs. To simplify the presentation, let me start by assuming that the input ERC matrix has a unique L per ERC. The contribution of the ith ERC $\mathbf{a}_i$ to the current ranking vector according to this re-ranking rule can thus be summarised with the corresponding UPDATE VECTOR $\overline{\mathbf{a}}_i$, as in (83): the entry corresponding to the loser-preferrer is equal to -1; the entries corresponding to winner-preferrers are set equal to the corresponding promotion amount $p > 0$; all other entries are 0.

(83)

$$\mathbf{a}_i = [a_1, \ldots, a_n] \rightarrow \overline{\mathbf{a}}_i = \begin{bmatrix} \overline{a}_1 \\ \vdots \\ \overline{a}_n \end{bmatrix} \text{where } \overline{a}_k = \begin{cases} p & \text{if } a_k = \text{W} \\ -1 & \text{if } a_k = \text{L} \\ 0 & \text{otherwise} \end{cases}$$

Suppose that the initial RVs are all null. The current ranking vector $\boldsymbol{\theta}^t$ entertained at time t by the EDRA with the re-ranking rule (82) can be described as in (84), i.e. as a combination of the update vectors, each multiplied by the number of updates α^t_i triggered by the corresponding ith ERC in the run considered up to time t. Equation (43), obtained in the discussion of Pater's counterexample in §5.3, is a special case of the general equation (84). Of course, the coefficients α^t_i are by definition all non-negative. Thus the identity (84) can be summarised by saying that the current ranking vector is a CONIC COMBINATION of the m update vectors.

(84)

$$\boldsymbol{\theta}^t = \alpha^t_i \overline{\mathbf{a}}_1 + \ldots + \overset{\substack{\text{number of updates} \\ \text{triggered by the } i\text{th} \\ \text{ERC up to time } t \\ \downarrow}}{\alpha^t_i} \underset{\substack{\uparrow \\ \text{update vector} \\ \text{corresponding to} \\ \text{the } i\text{th ERC}}}{\overline{\mathbf{a}}_i} + \ldots + \alpha^t_m \overline{\mathbf{a}}_m$$

As the current ranking vector is a conic combination of the update vectors, it is interesting to study the conic geometry of these vectors, i.e. the formal properties of their conic combinations. Here is a particularly important conic property. The update vectors are called CONICALLY INDEPENDENT provided that there are no coefficients $\alpha_1, \ldots, \alpha_m$ that satisfy the three conditions in (85); see Bertsekas *et al.* (2003). In other words, it is not possible to synthesise the null vector as a conic combination of the update vectors, unless of course the coefficients are all set equal to zero.

(85) a. $\alpha_1 \overline{\mathbf{a}}_1 + \ldots + \alpha_m \overline{\mathbf{a}}_m = \mathbf{0}$
b. $\alpha_i \geq 0$ for all $i = 1, \ldots, m$
c. $\alpha_i \neq 0$ for some $i = 1, \ldots, m$

Fact 6 below says that OT-consistency of the input ERC matrix (with a unique L per row) entails conic independence of the corresponding update vectors, and Fact 7 says that conic independence of the update vectors in turn entails that the EDRA cannot loop. Fact 5 thus follows from the two auxiliary Facts 6 and 7. The assumption that the input ERCs have a unique L per row can be easily dropped, as discussed at the end of this section.

FACT 6
Consider an input ERC matrix that has a unique L per row. If it is consistent, then the corresponding update vectors defined in (83) are conically independent.

Proof. Recall from Fact 1 that any consistent ERC matrix has the shape in (22) (repeated in (86)), modulo reordering of its rows and its columns and relabelling of the constraints. (86) has a top block of rows whose first entry is W, followed by a second block of rows whose first entry is *e* and whose second entry is W, and so on.

(86)
$$
\begin{array}{ccccccc|l}
\mathrm{C}_1 & \mathrm{C}_2 & \cdots & \cdots & \cdots & \cdots & \mathrm{C}_n & \\
\mathrm{W} & & & & & & & \\
| & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \text{1st block} \\
\mathrm{W} & & & & & & & \\
\hline
e & \mathrm{W} & & & & & & \\
| & | & \cdots & \cdots & \cdots & \cdots & \cdots & \text{2nd block} \\
e & \mathrm{W} & & & & & & \\
\hline
 & & \ddots & \cdots & \cdots & \cdots & \cdots & \vdots \\
\hline
e & e & - & e & \mathrm{W} & & & \\
| & | & & | & | & \cdots & \cdots & \text{final block} \\
e & e & - & e & \mathrm{W} & & &
\end{array}
$$

For consistency with standard notation from Linear Algebra, in (83) I paired up a *row* of the input ERC matrix with a corresponding *column* update vector. To get around this rows/columns mismatch, let me turn (86) upside down (i.e. transpose), so that rows become columns, as in (87).

(87)
$$\begin{array}{cccc} \text{1st block} & \text{2nd block} & & \text{final block} \end{array}$$
$$\left[\begin{array}{c|c|c|c} \mathrm{w}-\mathrm{w} & e\,-\,e & & e\,-\,e \\ \hline \cdots & \mathrm{w}-\mathrm{w} & & e\,-\,e \\ \hline & & \ddots & \\ \hline \cdots & \cdots\ - & & \mathrm{w}-\mathrm{w} \\ \hline \vdots & \vdots & \vdots & \vdots \end{array}\right]$$

The update vectors can now be read straightforwardly from (87): the *i*th update vector is obtained by looking at the *i*th column. Recall that the mapping (83) from ERCs into update vectors replaces a *e* with a 0 and a w with the positive quantity $p > 0$. The collection of update vectors can thus be made a little more explicit, as in (88). (Each update vector can have a different value for the promotion amount p. This fact does not play any role in the reasoning, so I do not encode it explicitly in the notation, and use the same p for all update vectors.)

(88)
$$\underbrace{\begin{bmatrix}p\\ \hline \\ \hline \\ \hline \\ \hline \\ \end{bmatrix}, \dots, \begin{bmatrix}p\\ \hline \\ \hline \\ \hline \\ \hline \\ \end{bmatrix}}_{\text{1st block}}, \underbrace{\begin{bmatrix}0\\ \hline p\\ \hline \\ \hline \\ \hline \\ \end{bmatrix}, \dots, \begin{bmatrix}0\\ \hline p\\ \hline \\ \hline \\ \hline \\ \end{bmatrix}}_{\text{2nd block}}, \dots, \underbrace{\begin{bmatrix}0\\ \hline |\\ \hline 0\\ \hline p\\ \hline \\ \end{bmatrix}, \dots, \begin{bmatrix}0\\ \hline |\\ \hline 0\\ \hline p\\ \hline \\ \end{bmatrix}}_{\text{final block}}$$

Suppose that a conic combination of these update vectors (88) with some non-negative coefficients yields the null vector, namely that conditions (85a) and (b) hold. Let's focus on the first component of this conic combination, as in (89). The first component of the update vectors in the 1st block is always positive (recall that $p > 0$ by hypothesis). Suppose there are k vectors in the 1st block. The first component of the remaining $m - k$ update vectors is always null. In order for the first component of this conic combination to be zero, the non-negative coefficients that multiply the update vectors in the 1st block must all be null, namely $\alpha_1 = \ldots = \alpha_k = 0$.

(89)
$$\underbrace{\alpha_1 \begin{bmatrix}p\\ \\ \\ \end{bmatrix} + \ldots + \alpha_k \begin{bmatrix}p\\ \\ \\ \end{bmatrix}}_{\text{1st block}} \underbrace{+ \alpha_{k+1} \begin{bmatrix}0\\ \\ \\ \end{bmatrix} + \ldots + \alpha_m \begin{bmatrix}0\\ \\ \\ \end{bmatrix}}_{\text{remaining blocks}} = \begin{bmatrix}0\\ \\ \\ \end{bmatrix}$$

As their coefficients are null, the update vectors in the 1st block can be ignored in the conic combination. By looking at the second component and reasoning analogously, I conclude that also the coefficients that multiply the update vectors in the 2nd block are null. By repeating the reasoning, I conclude that these multiplicative coefficients are all null, contradicting condition (85c) in the definition of conic independence. □

FACT 7

If the update vectors are conically independent, then the EDRA cannot loop back to a current ranking vector it had previously updated.

Proof. Suppose by contradiction that the EDRA can indeed loop back to a ranking vector that it had dismissed at a previous time. This means that it is possible for the algorithm to walk through a learning path with the properties in (90).

(90) a. The EDRA entertains the same ranking vector at two times, t and t'.
b. Assume for concreteness that time t precedes time t'.
c. The EDRA entertains a different ranking vector at a time between t and t'.

Assumption (90a), that the ranking vectors $\boldsymbol{\theta}^t$ and $\boldsymbol{\theta}^{t'}$ entertained at times t and t' coincide, can be expressed as the identity (91a), using the description (84) of the current ranking vector in terms of update vectors. Here, α_i^t and $\alpha_i^{t'}$ are the number of updates triggered by ERC 1 up to time t and t' respectively; an analogous interpretation holds for the other coefficients. As the number of updates grows with time, assumption (90b), that time t' follows time t, thus entails that the coefficient $\alpha_i^{t'}$ at time t' is larger than or equal to the corresponding coefficient α_i^t at time t, as stated in (91b). Furthermore, assumption (90c) entails that some update has happened at some time between t and t', so that at least one of the coefficients has increased by at least 1 from time t to time t', as stated in (91c).

(91) a. $\alpha_1^t\overline{\mathbf{a}}_1 + \ldots + \alpha_m^t\overline{\mathbf{a}}_m = \alpha_1^{t'}\overline{\mathbf{a}}_1 + \ldots + \alpha_m^{t'}\overline{\mathbf{a}}_m$
b. $\alpha_i^{t'} \geq \alpha_i^t$ for all $i = 1, \ldots, m$
c. $\alpha_i^{t'} \neq \alpha_i^t$ for some $i = 1, \ldots, m$

By moving everything to the right-hand side, (91a) can of course be restated as in (92a), where I have introduced the coefficients $\alpha_i = \alpha_i^{t'} - \alpha_i^t$ for all $i = 1, \ldots, m$. The property (91b), that $\alpha_i^{t'}$ is larger than or equal to α_i^t because time t' follows time t, can then be restated as (92b), that all coefficients α_i are non-negative. The property (91c), that some coefficient $\alpha_i^{t'}$ is different from the corresponding coefficient α_i^t because some update has happened between times t and t', can be restated as the property (92c), that at least one of the coefficients α_i is non-null.

(92) a. $\alpha_1\overline{\mathbf{a}}_1 + \ldots + \alpha_m\overline{\mathbf{a}}_m = 0$
b. $\alpha_i \geq 0$ for all $i = 1, \ldots, m$
c. $\alpha_i \neq 0$ for some $i = 1, \ldots, m$

The conditions in (92) say that the null vector can be synthesised as a conic combination of the update vectors, without the coefficients $\alpha_1, \ldots, \alpha_m$ all being null. This contradicts the hypothesis that the update vectors are conically independent. □

To conclude the proof of Fact 5, I need to consider the case where the input ERC matrix contains rows with multiple L's. The additional difficulty in this case is that the contribution of the ith ERC to the current ranking vector depends on the number of currently undominated loser-preferrers, i.e. it can be different at different times, and thus cannot be distilled into a unique update vector $\overline{\mathbf{a}}_i$, as in (83). But this difficulty can be straightforwardly overcome, at the expense of a slightly more cumbersome notation. Let m be the total number of ERCs. Suppose that the ith ERC has l_i loser-preferrers $C_{k_1}, \ldots, C_{k_{l_i}}$. Let $\mathbb{C}^i_1, \ldots, \mathbb{C}^i_{2^{l_i}-1}$ be all $2^{l_i}-1$ non-empty subsets of the set $\{C_{k_1}, \ldots, C_{k_{l_i}}\}$ of loser-preferrers. For every such subset $\mathbb{C}^i_j$, let $\overline{\mathbf{a}}_{i,j}$ be the update vector defined as follows: the components corresponding to the loser-preferrers in the subset $\mathbb{C}^i_j$ are equal to -1; the components corresponding to winner-preferrers are equal to p; the remaining components are equal to 0. Furthermore, let $\alpha^t_{i,j}$ be the number of updates triggered by this ith ERC up to time t, because all and only the loser-preferrers in the set $\mathbb{C}^i_j$ were currently undominated. The current ranking vector can then be expressed as a conic combination of these update vectors through these non-negative coefficients, namely $\boldsymbol{\theta}^t = \sum_{i=1}^{m} \sum_{j=1}^{2^{l_i}-1} \alpha^t_{i,j}\, \overline{\mathbf{a}}_{i,j}$. Again, these update vectors $\overline{\mathbf{a}}_{i,j}$ are conically independent. I can thus trivially extend the preceding reasoning.

Appendix D: On the number of updates for smallest non-calibrated promotion

Recall from §7.5 that the aggravated Pater's ERC matrix for n constraints is obtained from the corresponding diagonal matrix by adding two W's to the right of each L. To illustrate, I give in (93) the matrix corresponding to $n = 7$ constraints. It has $6 = n - 1$ ERCs; it has a W on every diagonal entry, followed by an L followed in turn by two more W's (except for the last two rows, whose L's are followed by one and zero W's respectively).

(93)

$$\begin{array}{r|ccccccc|}
 & C_1 & C_2 & C_3 & C_4 & C_5 & C_6 & C_7 \\
\text{ERC 1} & \text{W} & \text{L} & \text{W} & \text{W} & & & \\
\text{ERC 2} & & \text{W} & \text{L} & \text{W} & \text{W} & & \\
\text{ERC 3} & & & \text{W} & \text{L} & \text{W} & \text{W} & \\
\text{ERC 4} & & & & \text{W} & \text{L} & \text{W} & \text{W} \\
\text{ERC 5} & & & & & \text{W} & \text{L} & \text{W} \\
\text{ERC 6} & & & & & & \text{W} & \text{L}
\end{array}$$

Consider a run of the EDRA on the aggravated Pater's input ERC matrix for n constraints. Suppose the algorithm starts from null initial RVs, and that it uses the re-ranking rule (59) with the smallest non-calibrated promotion amount, repeated in (94) for the case of input ERCs with a single loser-preferrer, as in the case of aggravated Pater's ERC matrix.

(94) a. Decrease the RV of the loser-preferrer by 1.
b. Increase the RV of each of the w winner-preferrers by l/w.

The convergence Theorem 3 ensures that after a finite number of errors the EDRA will converge to a final ranking vector consistent with the input ERC matrix, and learning will cease. Yet the theorem does not provide any estimate of the number of errors made before convergence. This section shows that this number grows exponentially with the number n of constraints, as anticipated in §7.5.

For concreteness, suppose the input matrix is the aggravated Pater's matrix (93), corresponding to $n = 7$ constraints. Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_7)$ be the final ranking vector the EDRA has converged on. Let $\alpha_1, \ldots, \alpha_6$ be the total number of updates triggered by each of the six input ERCs in the run considered. The final RVs $\theta_1, \ldots, \theta_7$ can be expressed in terms of the coefficients $\alpha_1, \ldots, \alpha_6$, as in (95).

(95)
$$\begin{aligned}
\theta_1 &= \tfrac{1}{3}\alpha_1\\
\theta_2 &= -\alpha_1 + \tfrac{1}{3}\alpha_2\\
\theta_3 &= \tfrac{1}{3}\alpha_1 - \alpha_2 + \tfrac{1}{3}\alpha_3\\
\theta_4 &= \tfrac{1}{3}\alpha_1 + \tfrac{1}{3}\alpha_2 - \alpha_3 + \tfrac{1}{3}\alpha_4\\
\theta_5 &= \tfrac{1}{3}\alpha_2 + \tfrac{1}{3}\alpha_3 - \alpha_4 + \tfrac{1}{2}\alpha_5\\
\theta_6 &= \tfrac{1}{3}\alpha_3 + \tfrac{1}{3}\alpha_4 - \alpha_5 + \alpha_6\\
\theta_7 &= \tfrac{1}{3}\alpha_4 + \tfrac{1}{2}\alpha_5 - \alpha_6
\end{aligned}$$

Here is how these equations are obtained. The RV θ_1 of constraint C_1 starts out null. It is only modified when ERC 1 triggers an update, in which case, it is increased by 1/3, as ERC 1 contains 3 winner-preferrers. In other words, the final RV θ_1 of constraint C_1 is 1/3 of the total number α_1 of updates triggered by ERC 1, as stated by the first equation in (95). Analogously, the RV θ_2 of constraint C_2 starts out null, is decreased by 1 every time ERC 1 triggers an update and is increased by 1/3 every time ERC 2 triggers an update, whereby we get the second equation in (95). The remaining equations in (95) are obtained analogously.

The input matrix (93) is only consistent with the ranking $C_1 \gg C_2 \ldots \gg C_7$. As the final ranking vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_7)$ entertained by the EDRA at convergence is consistent with the input matrix of ERCs, it must therefore satisfy the six strict inequalities $\theta_1 > \theta_2, \ldots, \theta_6 > \theta_7$. Consider for instance the first of these six inequalities, repeated in (96a). Using the first two equations in (95), this inequality can be rewritten as in (96b), in terms of the numbers of updates α_1 and α_2 triggered by ERC 1 and ERC 2 respectively. If both sides of inequality (96b) are multiplied by the constant 3, we get the equivalent inequality (96c). As the variables α_1, α_2, as well as the coefficients, are integers, the strict inequality (96c) is equivalent to the loose inequality (96d), where I have added 1 to the right-hand side.

(96) a. $\theta_1 > \theta_2$

b. $\frac{1}{3}\alpha_1 > -\alpha_1 + \frac{1}{3}\alpha_2$

c. $\alpha_1 > -3\alpha_1 + \alpha_2$

d. $\alpha_1 \geq -3\alpha_1 + \alpha_2 + 1$

By reasoning this way, I conclude that the six strict inequalities $\theta_1 > \theta_2, \ldots, \theta_6 > \theta_7$ are equivalent to the six inequalities in (97) in terms of the number of updates $\alpha_1, \ldots, \alpha_6$ triggered by ERC 1 through ERC 6 respectively.

(97)
$$\begin{aligned}
\theta_1 > \theta_2 &\Leftrightarrow & \alpha_1 &\geq 1 - 3\alpha_1 + \alpha_2\\
\theta_2 > \theta_3 &\Leftrightarrow & -3\alpha_1 + \alpha_2 &\geq 1 + \alpha_1 - 3\alpha_2 + \alpha_3\\
\theta_3 > \theta_4 &\Leftrightarrow & \alpha_1 - 3\alpha_2 + \alpha_3 &\geq 1 + \alpha_1 + \alpha_2 - 3\alpha_3 + \alpha_4\\
\theta_4 > \theta_5 &\Leftrightarrow & 2\alpha_1 + 2\alpha_2 - 6\alpha_3 + 2\alpha_4 &\geq 1 + 2\alpha_2 + 2\alpha_3 - 6\alpha_4 + 3\alpha_5\\
\theta_5 > \theta_6 &\Leftrightarrow & 2\alpha_2 + 2\alpha_3 - 6\alpha_4 + 3\alpha_5 &\geq 1 + 2\alpha_3 + 2\alpha_4 - 6\alpha_5 + 6\alpha_6\\
\theta_6 > \theta_7 &\Leftrightarrow & 2\alpha_3 + 2\alpha_4 - 6\alpha_5 + 6\alpha_6 &\geq 1 + 2\alpha_4 + 3\alpha_5 - 6\alpha_6
\end{aligned}$$

The total number of updates performed by the EDRA in the run considered coincides with the sum $\alpha_1 + \ldots + \alpha_6$ of the number α_1 of updates triggered by ERC 1 plus the number α_2 of updates triggered by ERC 2 and so on down to the number α_6 of updates triggered by ERC 6. Furthermore, these non-negative numbers $\alpha_1, \ldots, \alpha_6$ must satisfy the inequalities in (97). Thus the number of updates performed by the EDRA to reach convergence cannot be smaller than the solution of the optimisation problem (98). In other words, the solution of this optimisation problem provides a bound on the best-case number of updates performed by the EDRA on the input matrix (93). As (98) is a linear program, it can be easily solved with standard linear programming techniques.

(98) minimise: $\alpha_1 + \ldots + \alpha_6$
subject to: $\alpha_1, \ldots, \alpha_6$ satisfy the inequalities in (97)
$\alpha_1, \ldots, \alpha_6 \geq 0$

The reasoning just developed in the concrete case of the aggravated Pater's matrix in (93) corresponding to $n = 7$ constraints extends to the case of an arbitrary number n of constraints. I can always construct an optimisation problem akin to (98) that provides a bound on the best-case number of updates performed by the EDRA on that aggravated Pater's matrix. The solution of the optimisation problems thus obtained for the aggravated Pater's matrices corresponding to various choices of the number n of constraints are reported in Table Ia. (These values were computed using the Matlab file MinimumRunningTime.m. It takes as input the aggravated Pater's ERC matrix corresponding to n constraints, for any n. It constructs the corresponding optimisation problem akin to (98), generalising the reasoning just presented here in the special case of the aggravated Pater's comparative tableaux corresponding to $n = 7$ constraints, and it solves this optimisation problem using Matlab built-in subroutines for linear programming. Aggravated Pater's comparative matrices for $n = 5, 7, 9, 11, 13, 15$ constraints are provided in the file AggravatedPaterMatrices.txt. Both files are available here.)

Let me close by pointing out the close parallelism between the reasoning presented in this section and the explanation for Pater's (2008) counterexample against the GLA's convergence provided in §5.3. The equations in (95) are analogous to those in (44) in §5.3, and both are a special case of the vector equation (84) in Appendix C. The inequalities in (97) are analogous to those in (45) in §5.3. Finally, showing that there are no coefficient α's that solve the inequalities in (97) and add up to a small number corresponds to the final step of the explanation of Pater's counterexample, which showed that there are no coefficient α's that solve the inequalities (45).

Appendix E: On the notion of OT-consistency for ranking vectors

In §2.3, I defined OT-consistency between an ERC and a ranking vector as in (14), repeated below in (99). This appendix qualifies this definition with some remarks.

(99) A ranking vector is (OT-)consistent with an ERC provided that each of its refinements is consistent with that ERC, according to the original notion of consistency (9).

1 Comparison with Tesar & Smolensky's (1998) notion of OT-consistency for stratified hierarchies

T&S introduce a variant of the standard OT framework, replacing total rankings with stratified hierarchies, which can assign multiple constraints to a single stratum. Stratified hierarchies are thus equivalent to ranking vectors, which can assign the same ranking values to multiple constraints. Yet the notion of consistency assumed by T&S is very different from the one in (99), as T&S allow for the tie among constraints assigned to the same stratum to be resolved additively. Without getting into the details of this alternative definition of OT-consistency, let me illustrate the difference with an example. Consider the ERC (100a) together with the ranking vector (100b). Because of the two identical ranking values $\theta_1 = \theta_2 = 2$, this ranking vector would represent a stratified hierarchy that assigns both C_1 and C_2 to the top stratum, with C_3 ranked underneath.

(100) a. $\begin{array}{cccc} & \mathrm{C}_1 & \mathrm{C}_2 & \mathrm{C}_3 \\ \text{ERC} & [\ \text{W} & \text{L} & \text{W}\ ] \end{array}$ b. $\begin{array}{cccc} & \mathrm{C}_1 & \mathrm{C}_2 & \mathrm{C}_3 \\ \boldsymbol{\theta} = & (\ 2 & 2 & 1\ ) \end{array}$

According to the alternative definition of OT-consistency introduced by T&S, the ranking vector (100b) is indeed consistent with the ERC (100a), as the L and the W of the two equally highest-ranked constraints C_1 and C_2 'cancel out' because of the additive interaction between equally rankedconstraints. But the ranking vector (100b) is not consistent with the ERC (100a) according to the definition (99), since the ranking vector (100b) admits the refinement $C_2 \gg C_1 \gg C_3$, which is not consistent with the ERC (100a). As the paper assumes the classical notion of OT-consistency (99), it is framed squarely within standard OT. Contrary to what suggested by T&S, there is no need to step outside of the standard OT framework for algorithmic purposes.

2 Computational efficiency

A reviewer worries that the notion of OT consistency (99), which requires consistency to hold for *all* refinements, might not be efficiently computable. For instance, if all ranking values are identical, then don't we have to check consistency for all n! refinements, causing a complexity explosion? That is not the case. Here is a way to see this. Let $W(\mathbf{a})$ and $L(\mathbf{a})$ be the sets of winner- and loser-preferring constraints relative to an ERC $\mathbf{a}$. It turns out that a ranking vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ is consistent with an ERC $\mathbf{a}$ according to condition (99) provided the following strict inequality (101) holds, which says that the largest ranking value over winner-preferrers is larger than the largest ranking value over loser-preferrers.

(101) $$\max_{k \in W(\mathbf{a})} \theta_k > \max_{h \in L(\mathbf{a})} \theta_h$$

Furthermore, the inequality (101) can be checked in n steps (where n is the number of constraints). In the end, the consistency condition (99) can thus be efficiently computed.

Let me explain why the consistency condition (99) is equivalent to the strict inequality (101). Suppose that the latter inequality (101) holds. Every refinement of this ranking vector will then rank the winner-preferrer that attains the maximum $\max_{k \in W(\mathbf{a})} \theta_k$ above the loser-preferrer that attains the maximum $\max_{h \in L(\mathbf{a})} \theta_h$. In other words, it will rank this winner-preferrer above every loser-preferrer. Every refinement is thus consistent with the ERC $\mathbf{a}$, and condition (99) holds. *Vice versa*, suppose that inequality (101) does not hold, i.e. the largest ranking value over winner-preferrers is at most as large as the largest ranking value over loser-preferrers. Thus the current ranking vector admits a refinement that ranks the loser-preferrer that attains the maximum $\max_{h \in L(\mathbf{a})} \theta_h$ above the winner-preferrer that attains the maximum $\max_{k \in W(\mathbf{a})} \theta_k$. In other words, it admits a refinement that ranks a loser-preferrer above every winner-preferrer. This refinement is thus not consistent with the ERC $\mathbf{a}$, and condition (99) fails.

Finally, let me explain why inequality (101) can be checked in n steps (where n is the number of constraints). Posit $W = -\infty$ and $L = -\infty$. Scan through the current ranking values, for $k = 1, 2, \ldots n$. If θ_k is larger than

W (larger than L) and C_k is winner-preferring (loser-preferring), then set $W = \theta_k$ (set $L = \theta_k$). After having scanned all ranking values, condition (101) holds iff $W > L$. This process only takes n steps.

3 Consistency with Pater's ERC matrix

In §5.2, I considered Pater's (2008) ERC matrix (42). This ERC matrix requires the ranking $C_1 \gg C_2 \gg C_3 \gg C_4 \gg C_5$. Thus, a ranking vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$ is OT-consistent with this ERC matrix according to condition (99), provided it admits only that ranking as a refinement. This means in turn that the ranking values must satisfy the four strict inequalities $\theta_1 > \theta_2$, $\theta_2 > \theta_3$, $\theta_3 > \theta_4$ and $\theta_4 > \theta_5$. In §5.3, I showed that these four strict inequalities can be rewritten as in (45) in terms of the number of updates $\alpha_1, \ldots, \alpha_4$, triggered by the four ERCs in Pater's ERC matrix. Furthermore, I have claimed that these inequalities (45) admit no solution. To see this, rewrite these inequalities (45) as in (102), by moving everything on one side.

$$
(102)\quad \begin{array}{rrrrl}
+2\alpha_1 & -\alpha_2 & & & > 0 \\
-2\alpha_1 & +2\alpha_2 & -\alpha_3 & & > 0 \\
+\alpha_1 & -2\alpha_2 & +2\alpha_3 & -\alpha_4 & > 0 \\
 & +\alpha_2 & -2\alpha_3 & +2\alpha_4 & > 0
\end{array}
$$

If we sum all four inequalities together, we obtain inequality (103a). If we sum together only the second and third inequalities in (102), we obtain inequality (103b).

(103) a. $\alpha_1 - \alpha_3 + \alpha_4 > 0$

b. $-\alpha_1 + \alpha_3 - \alpha_4 > 0$

As the two inequalities thus derived in (103) are inconsistent, the four original inequalities (102) admit no solution.

ADDITIONAL REFERENCE

Bertsekas, Dimitri P., Angelia Nedic & Asuman E. Ozdaglar (2003). *Convex analysis and optimization*. Nashua, NH: Athena Scientific.