2020. *Proc Ling Soc Amer* 5(1). 144–149. https://doi.org/10.3765/plsa.v5i1.4694.

# From experiment results to a constraint hierarchy with the 'Rank Centrality' algorithm

Jennifer L. Smith[*]

**Abstract**. Rank Centrality (RC; Negahban, Oh, & Shah 2017) is a rank-aggregation algorithm that computes a total ranking of elements from noisy pairwise ranking information. I test RC as an alternative to incremental error-driven learning algorithms such as GLA-MaxEnt (Boersma & Hayes 2001; Jäger 2007) for modeling a constraint hierarchy on the basis of two-alternative forced-choice experiment results. For the case study examined here, RC agrees well with GLA-MaxEnt on the ordering of the constraints, but differs somewhat on the distance between constraints; in particular, RC assigns more extreme (low) positions to constraints at the bottom of the hierarchy than GLA-MaxEnt does. Overall, these initial results are promising, and RC merits further investigation as a constraint-ranking method in experimental linguistics.

**Keywords**. Constraint ranking algorithms; Rank Centrality; Maximum Entropy; experimental phonology; loanword phonology

**1. Introduction**. One way of testing hypotheses about linguistic competence is to collect judgment data in an experiment. However, interpreting the results is not always straightforward. Consider an experiment whose results are a set of domination decisions about pairs of constraints $C_i$, $C_j$ sampled from $\{C_1, ...C_n\}$. How can we extract an overall rank/weight hierarchy for $\{C_1, ...C_n\}$ based only on the proportion of $C_i \gg C_j$ responses for each $C_i$, $C_j$ pair—especially if each actual $C_i$, $C_j$ domination relation has the potential to be variable? I evaluate the ability of the general-purpose rank-aggregation algorithm Rank Centrality (RC; Negahban et al. 2017) to do just this, and I show that the RC results are promisingly similar to those of the state-of-the-art Gradual Learning Algorithm (GLA; Boersma & Hayes 2001).

RC was developed to model both the ordering of, and the distance between, items in a set, given data from comparisons between pairs of items. This ranking algorithm is designed to be computationally simple, to require *only* pairwise comparison data (i.e., how many times is *i* chosen over *j* out of all *i/j* comparisons?) rather than having explicit scores assigned *a priori* to items in the set, and to perform at least as well as existing algorithms (Negahban et al. 2017: §1).

Given these properties, RC is appealing as a potential means of interpreting phonological experiment results from a two-alternative forced-choice task—in which participants choose which of two linguistic forms they prefer—where the aim is to determine an overall constraint hierarchy. This paper tests the use of RC in such a context by applying it to the results of an experiment on Japanese nonce-loanword nativization (Smith & Tashiro 2019). The resulting constraint hierarchy for each participant is compared with that generated by a Maximum Entropy learner under the GLA (GLA-MaxEnt; Jäger 2007).

Section 2 first summarizes the test-case experiment, after which the GLA and RC algorithms are described in sections 3 and 4 respectively. The RC and GLA-MaxEnt constraint hierarchies

[*] The Japanese nonce-loan experiment was conducted in collaboration with Yuka Tashiro and was supported by a Schwab Academic Excellence grant from the Institute for the Arts and Humanities at UNC Chapel Hill. Thanks to Eric Bakovic, Ellen Broselow, Jeff Heinz, Larry Hyman, and Elliott Moreton for comments and discussion; particular thanks to Elliott Moreton for intensive discussion of graph theory and linear algebra. Author: Jennifer L. Smith, University of North Carolina at Chapel Hill (jlsmith@email.unc.edu).

derived from the same set of experiment results are compared in section 5, and conclusions and implications are discussed in section 6.

**2. Test-case experiment**. The forced-choice experiment results to be used for the comparison between the Rank Centrality and GLA-MaxEnt ranking algorithms are those from the study described in Smith & Tashiro (2019). This experiment investigated whether Japanese speakers have a synchronically productive markedness hierarchy for loan nativizations, such that certain ‘foreign’ characteristics are seen as more important to nativize than others. For discussion of the theoretical context of this study and its implications, see Smith & Tashiro (2019); the focus here is specifically on methods of determining a ranking from each participant’s raw results.

The markedness constraints under investigation were the following (adapted from Ito & Mester 1999), all of which are observed to drive phonological alternations in at least some lexical classes in Japanese.

(1) Markedness constraints tested in the nonce-loan nativization experiment

NoSI: Assign one * for every sequence of anterior coronal fricative + high front vowel

NoTI: Assign one * for every sequence of anterior coronal plosive + high front vowel

NoDD: Assign one * for every voiced geminate obstruent

NoNT: Assign one * for every nasal + voiceless obstruent sequence
(Hayes 1999; Pater 2001)

NoP: Assign one * for every singleton (non-geminate) [p]

Each trial compared two of the constraints in (1), as follows. Participants were presented with an English-like nonce word, such as *siftant* [**sɪ**ftæ**nt**], whose most faithful adaptation would violate both constraints in the pair, here NoSI and NoNT. Two Japanese nativizations were also presented, in this case [**si**ɸɯta***nd***o] and [***ɕi***ɸɯta**nt**o], where each incurred one of the constraint violations but *avoided* the other (by means of a nativization strategy attested in actual Japanese lexemes). Stimuli were presented both as audio and in the appropriate orthography, i.e., in the Roman alphabet for the English-like nonce form and in the *katakana* syllabary for the nativization options. The task was to choose which of the nonce loanword adaptations was ‘more natural’. In this example, a participant who chooses [***ɕi***ɸɯta**nt**o] rather than [**si**ɸɯta***nd***o] has chosen to satisfy NoSI at the expense of NoNT; this response is therefore compatible with the pairwise constraint ranking NoSI » NoNT.

Because each participant was presented with four nonce loans to test each constraint pair, it is possible to calculate a *score* for each constraint-pair ordering (e.g., NoSI » NoNT) for each participant, defined as the proportion of compatible nonce-loan nativization responses: 0, 0.25, 0.5, 0.75, or 1. But then, in order to interpret the results of the experiment, it is necessary to compute the *overall markedness hierarchy* for each participant. The goal of the original study was to compare these hierarchies across participants in order to address theoretical questions about the productivity of certain markedness-constraint domination relationships proposed in previous literature (e.g., Ito & Mester 1999; see Smith & Tashiro 2019 for discussion). Here, the focus is methodological: How can the pairwise constraint scores for a given participant be combined into an overall hierarchy including all five constraints?

If every participant had scores of 1 and 0 for each constraint pair—that is, strict constraint domination—then this task would be trivial, assuming of course that the hierarchy for each individual participant was consistent with the principle of transitivity (such that, given $C_i$ » $C_j$ and $C_j$ » $C_k$, we also find $C_i$ » $C_k$). However, the constraint-pair scores in the experiment results are in many cases not strict, but rather probabilistic; that is, $C_i$ » $C_j$ for 1/4, 2/4, or 3/4 trials. This makes determining the overall hierarchy across all five constraints less straightforward.

**3. The Gradual Learning Algorithm**. The state-of-the-art method for computing a constraint ranking or weighting from probabilistic response patterns of this type is to feed the output probability distributions to a learning algorithm, such as the Gradual Learning Algorithm (GLA; Boersma & Hayes 2001) as implemented in Praat (Boersma & Weenink 2019). The GLA returns a set of ranking values or weights for each constraint, which establishes both the *ordering* among the constraints and also the *distances* between them. Crucially, the GLA can learn a grammar with variable outputs, such that the algorithm's final-state grammar produces the variable outputs in the proportions in which they appear in the training data. The closer together the values of two constraints, the more likely their domination relationship is to vary when the grammar is invoked to produce an output form. This produces variation between the output favored by one of the closely ranked constraints and that favored by the other.

The GLA is an *incremental, error-driven* learner, so using this algorithm to model a grammar involves setting the relevant constraints at some initial ranking value/weight and then exposing the learner to data from the target language. On each learning trial (exposure to one piece of learning data), if the learner's current grammar produces the target output, the constraint values remain unchanged. However, if the output of the current grammar is not the target output, then all constraints favoring the target output have their values incrementally increased, and all constraints favoring the current winner have their values incrementally decreased.

As an incremental learner, the GLA makes testable predictions about learning paths in actual human-language acquisition, which is a strong point in its favor *as a model of learning* (see Jäger 2007 for discussion). The question of interest here, however, is: When the goal is not to model the course of human-language acquisition, but rather to interpret the results of a forced-choice experiment designed to uncover domination relationships between constraints in existing adult grammars, could RC be a viable, one-step alternative to the incremental, error-driven GLA?

**4. Rank Centrality**. The RC algorithm is applied as follows (Negahban et al. 2017: 271). The elements in the set to be compared (here, the five constraints tested in the experiment) are represented as nodes in a *directed graph*, and nodes *i* and *j* are connected by an edge $E_{ij}$ if elements *i* and *j* have been compared. Each edge $E_{ij}$ then has a weight that corresponds to the proportion of times node *j* is chosen over node *i* (in this case, by a single participant in the experiment); note that both *i* and *j* range over all elements in the set, and $E_{CmCn} + E_{CnCm} = 1$ for any specific pair of constraints $C_m$, $C_n$. From this directed graph, a *transition matrix* is computed as described by Negahban et al. 2017: §2.2), representing a random walk on the graph; again, the probability of moving from node *i* to node *j* in the random walk is determined by the proportion of times node *j* was chosen over node *i* by the participant under analysis. Finally, the stationary distribution of the graph is determined. Conceptually, this is the result of applying the transition matrix repeatedly, and represents the proportion of times that the random walk visits each node—i.e., how 'attractive' each node is compared to the others. Mathematically, the stationary distribution is the *largest left eigenvector* of the transition matrix. Negahban et al. (2017: 271) summarize the intuition behind RC as follows: "an object receives a high rank if it has been preferred to other high[-]ranking objects or if it has been preferred to many objects."

RC thus assigns a value, or weight, to each node, $0 \leq w \leq 1$, which represents both a *rank order* and a *distance* between elements in the graph. This is analogous to the output of the GLA, as described in §3 above. The following section now compares results from RC and the GLA applied to the same data set—the pairwise comparison results from the experiment described in Smith & Tashiro (2019). Are the rank orders and the distances between constraints derived by the two algorithms equivalent?

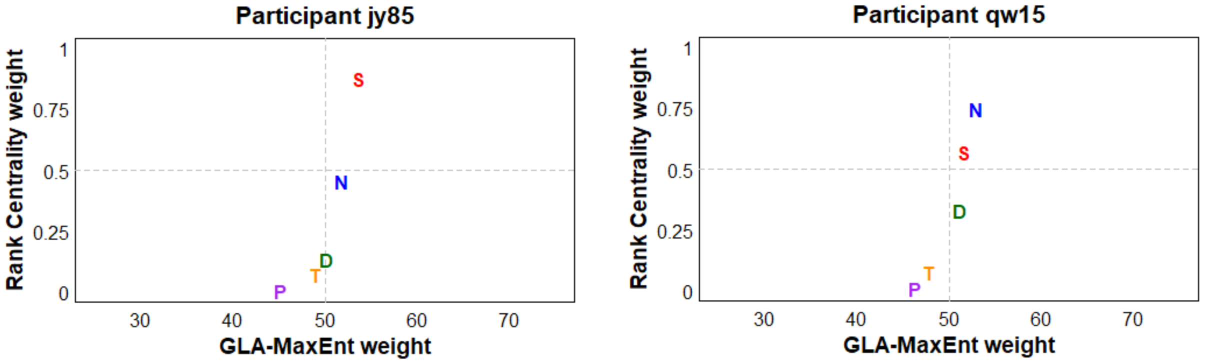
**5. Comparing RC and the GLA**. For each participant in the experiment, both RC and the GLA in its Maximum Entropy version (Jäger 2007) were applied to the $C_i$ » $C_j$ response proportions for all constraint pairs $C_i$, $C_j$. The MaxEnt version of the GLA was used, as opposed to other weighted or ranked constraint grammar models, because RC is based on a multinomial logit model (Negahban et al. 2017: §1), which is equivalent to a maximum-entropy model.

5.1 Methodology. The RC algorithm was applied to a directed graph representing all $C_i$ » $C_j$ response proportions for each participant. The output of the algorithm was a set of constraint weights for that participant, necessarily $0 \leq w \leq 1$.

The GLA-MaxEnt learner implemented in Praat (version 6.0.56; Boersma & Weenink 2019) was also applied to each participant's data, as follows. The Initial State grammar had all constraint weights set arbitrarily at 50. The Pair Distribution file, which represents the target-language learning data to which the learner will be exposed, was constructed to model the $Ci$ » $Cj$ response proportions for all constraint pairs $C_i$, $C_j$. The output of the algorithm was a set of constraint weights per participant, which turned out to fall in the range $25 \leq w \leq 75$.

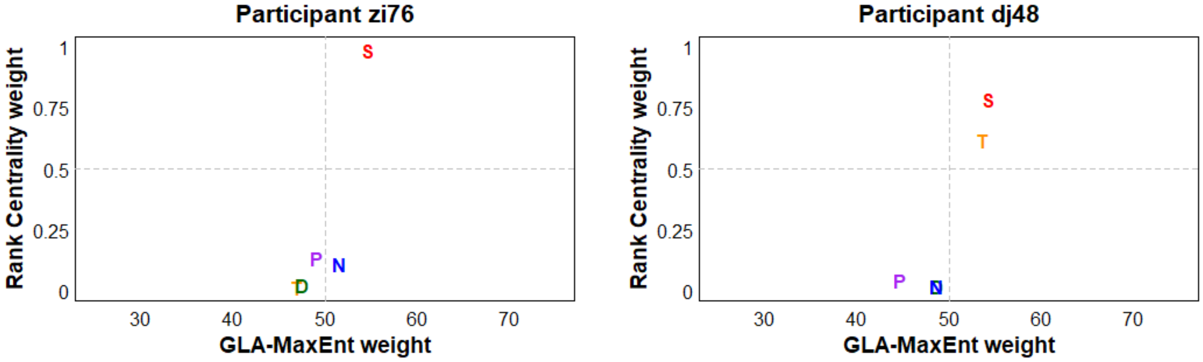
In order to compare the grammars produced by the two ranking algorithms, scatterplots were made for each participant, with RC constraint weights plotted against GLA-MaxEnt constraint weights; plots for two representative participants are shown in (2). Constraint names in plots are abbreviated as follows: **S**=NoSI, **T**=NoTI, **D**=NoDD, **N**=NoNT, **P**=NoP.

(2) Comparison plots for two representative participants



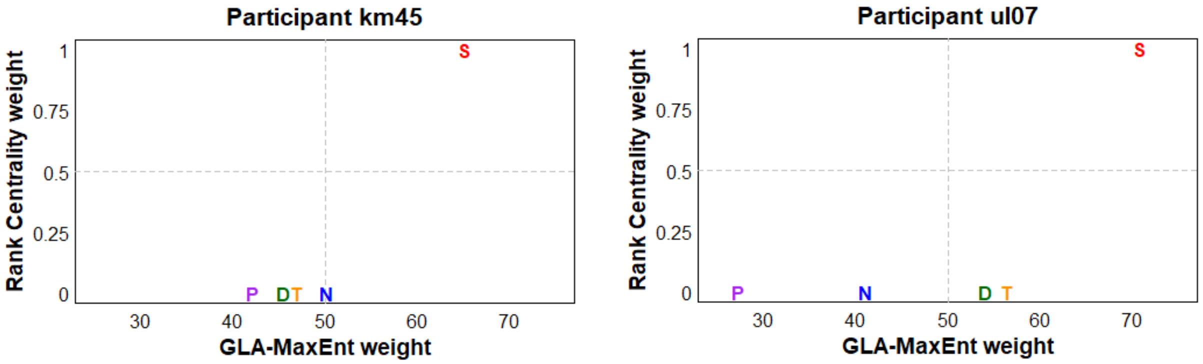
5.2 Results. The first comparison made was to determine whether RC and GLA derive the same *rank order* for the constraints. In most cases, the order is indeed consistent. In 11/40 participant grammars, two constraints show the reverse relative order under the two ranking algorithms. However, in all such reversals, the constraint values are in fact very close together, suggesting that in practice, the grammar would likely show a good deal of variability between outputs favored by each of the constraints in question. The plots for the two participants with the most extreme constraint reversals are shown in (3), with a reversal between NoP and NoNT for participant zi76, and a reversal between NoP and {NoDD, NoNT} for participant dj48; even in these most extreme cases, the numerical differences between the values of the 'reversing' constraints are very small.

(3) The two participants with the most extreme constraint reversals



The second comparison made was to determine whether RC and GLA-MaxEnt derive equivalent *distances* between constraints. This time, the match between the two ranking algorithms is not so close, especially at the low end of the range of values. Overall, most of the plots are S-shaped (as seen in (2) and (3)); that is, RC tends to make the ends of the range, especially the low end, more polarized than GLA-MaxEnt does. This effect is most salient for 14 participants whose grammars have NoSI weighted relatively far above the other constraints, as exemplified in (4). For these cases, RC simply assigns a weight of 1 to NoSI and 0 to all other constraints, but GLA-MaxEnt makes more distinctions among the lower constraints.

(4) Polarized RC results at the low end of the range



**6. Discussion and conclusions**. In summary, the results for *rank order* are very promising: RC ordered the constraints in much the same way as GLA-MaxEnt, except in cases where the constraints are very close together and therefore some variable behavior is expected. The results for *distance*, however, are not so similar. In particular, RC seems to show polarization of values at the low end of the range.

A possible reason for this low-end effect is that RC works best with large numbers of *i–j* comparisons; there may be too few comparisons in this set of experiment results. A variant of RC, known as *Regularized RC* (Negahban et al. 2017: 273), includes a prior probability term to compensate for small numbers of observations. Regularized RC, given an appropriate prior, might prove to be a better match for GLA-MaxEnt in interpreting these kinds of two-alternative forced-choice experiment results.

## References

Boersma, Paul & Bruce Hayes. 2001. Empirical tests of the Gradual Learning Algorithm. *Linguistic Inquiry* 32(1). 45–86. https://doi.org/10.1162/002438901554586.

Boersma, Paul & David Weenik. 2019. Praat, version 6.0.56, retrieved June 24, 2019. http://www.praat.org/.

Ito, Junko & Armin Mester. 1999. The structure of the phonological lexicon. In Natsuko Tsujimura (ed.), *The handbook of Japanese linguistics*. 62–100. Malden, MA: Blackwell.

Jäger, Gerhard. 2007. Maximum entropy models and stochastic Optimality Theory. In Annie Zaenen et al. (eds.), *Architectures, rules, and preferences: Variations on themes by Joan W. Bresnan*. 467–479. Stanford: CSLI. http://www.sfs.uni-tuebingen.de/~gjaeger/publications/mesga.pdf.

Negahban, Sahand, Sewoong Oh & Devavrat Shah. 2017. Rank Centrality: Ranking from pairwise comparisons. *Operations Research* 65(1). 266–287. https://doi.org/10.1287/opre.2016.1534.

Smith, Jennifer L. & Yuka Tashiro. 2019. Nonce-loan judgments and impossible-nativization effects in Japanese. *Proceedings of the Linguistic Society of America (PLSA)* 4. 26:1-14. https://doi.org/10.3765/plsa.v4i1.4531.