

# Invited Talk: Counting Rankings

**Jason Riggle**  
University of Chicago  
jriggle@uchicago.edu

## Abstract

In this talk, I present a recursive algorithm to calculate the *number* of rankings that are consistent with a set of data (optimal candidates) in the framework of Optimality Theory (OT; Prince and Smolensky 1993).<sup>1</sup> Computing this quantity, which I call *r*-volume, makes possible a simple and effective Bayesian heuristic in learning – *all else equal, choose candidates that are preferred by the highest number of rankings consistent with previous observations*. This heuristic yields an *r*-volume learning algorithm (RVL) that is guaranteed to make fewer than  $k \lg k$  errors while learning rankings of  $k$  constraints. This log-linear error bound is an improvement over the quadratic bound of Recursive Constraint Demotion (RCD; Tesar and Smolensky 1996) and it is within a logarithmic factor of the best possible mistake bound for any OT learning algorithm.

**Computing *r*-volume:** The violations in an OT tableau can be given as a  $[n \times k]$  array of integers in which the first row  $t_1$  corresponds to the winner. Following Prince (2002), the ranking information can be extracted by comparing  $t_1$  with each ‘losing’ row  $t_2, \dots, t_n$  to create an Elementary Ranking Condition as follows:  $erc(t_1, t_i) = \langle \alpha_1, \dots, \alpha_k \rangle$  where  $\alpha_j = \mathbf{L}$  if  $t_{1,j} < t_{i,j}$ ,  $\alpha_j = \mathbf{W}$  if  $t_{1,j} > t_{i,j}$ , and  $\alpha_j = \mathbf{e}$  otherwise. The meaning of  $\alpha$  is that at least one constraint associated with  $\mathbf{w}$  dominates all those associated with  $\mathbf{L}$ .

<i>input</i>	$\mathbb{C}_1$	$\mathbb{C}_2$	$\mathbb{C}_3$	
<i>candidate</i> $t_1$	*	**		<i>winner</i>
<i>candidate</i> $t_2$	**	*		$erc(t_1, t_2) = \langle \mathbf{W}, \mathbf{L}, \mathbf{e} \rangle$ i.e. $t_1$ beats $t_2$ if $\mathbb{C}_1$ outranks $\mathbb{C}_2$
<i>candidate</i> $t_3$			**	$erc(t_1, t_3) = \langle \mathbf{L}, \mathbf{L}, \mathbf{W} \rangle$ i.e. $t_1$ beats $t_3$ if $\mathbb{C}_3$ outranks $\mathbb{C}_1$ and $\mathbb{C}_2$
<i>candidate</i> $t_4$		***	*	$erc(t_1, t_4) = \langle \mathbf{L}, \mathbf{W}, \mathbf{W} \rangle$ i.e. $t_1$ beats $t_4$ if $\mathbb{C}_2$ or $\mathbb{C}_3$ outranks $\mathbb{C}_1$

For a set  $E$  of length- $k$  ERCs,  $E-w_i$  denotes a set  $E'$  derived from  $E$  by removing ERCs with  $\mathbf{w}$  in column  $i$  and removing column  $i$ .

$$r\text{-vol}(E^k) = \sum_{1 \leq i \leq k} \begin{cases} 0 & \text{if } x_i = \mathbf{L} \text{ for any } x \in E \\ (k-1)! & \text{if } x_i = \mathbf{W} \text{ for all } x \in E \\ r(E-w_i) & \text{otherwise} \end{cases}$$

**Mistake bounds:** To make predictions, RVL selects in each tableau the candidate that yields the highest *r*-volume when the ERCs that allow it to win are combined with  $E$  (the ERCs for past winners). To establish a mistake bound, assume that the RVL chooses candidate  $e$  when, in fact, candidate  $o$  was optimal according to the target ranking  $\mathcal{R}_T$ . Assuming  $e \neq o$ , the rankings that make  $o$  optimal must be half or fewer of the rankings consistent with  $E$  or else RVL would have chosen  $o$ . Because all rankings that make candidates other than  $o$  optimal will be eliminated once the ERCs for  $o$  are added to  $E$ , each error reduces the number of rankings consistent with all observed data by at least half and thus there can be no more than  $\lg k!$  errors.

**Applications:** The *r*-volume seems to encode ‘restrictiveness’ in a way similar to Tesar and Prince’s (1999) *r*-measure. As a factor in learning, it predicts typological frequency (cf. Bane and Riggle 2008) and priors other than the ‘flat’ distribution over rankings can easily be included to test models of ranking bias. More generally, this research suggests the concept of *g*-volume for any parameterized model of grammar.

<sup>1</sup>Full bibliography available on the Rutgers Optimality Archive (roa.rutgers.edu) with the paper Counting Rankings.