HG has no computational advantages over OT: consequences for the theory of OT online algorithms

Giorgio Magri

October 28, 2010

Abstract — Various authors have recently endorsed Harmonic Grammar (HG) as a replacement of Optimality Theory (OT). One argument for this move is based on computational considerations: OT looks prima facie like an exotic framework with no correspondent in Machine Learning, and the replacement with HG allows methods and results form Machine Learning to be imported within Computational Phonology; see for instance Potts et al. (2010), Pater (2009), Hayes and Wilson (2008), Coetzee and Pater (2008), Boersma and Pater (2007, 2008), Jesney and Tessier (2007, 2008), among others. This paper shows that this argument in favor of HG and against OT is wrong: I prove a simple, general result that says that algorithms for HG can be rather trivially adapted to OT. Thus, HG has no computational advantages over OT. This simple result has far reaching implications for Computational OT, as it allows classical methods and techniques from Machine Learning to be imported within Computational OT. I illustrate the fruitfulness of this new approach to Computational OT by showing that it leads to substantial progress in the theory of online algorithms for OT. In particular, I show that it leads to a convergence proof for a slight variant of Boersma's (1997) (non-stochastic) Gradual Learning Algorithm, based on convergence for the classical Perceptron Algorithm.

1 Introduction

The peculiar property of *Optimality Theory* (henceforth: OT) is that it uses *constraint ranking* and thus enforces *strict domination*, according to which the highest ranked relevant constraint "takes it all"; see Prince and Smolensky (2004). Because of this property, OT looks *prima facie* like a rather exotic combinatorial framework. Exotic in the sense that it does not seem to have any close correspondent within core Machine Learning.[1] For this reason, Computational OT has been developed in the current literature along the lines described in (1). Tesar and Smolensky (1998) well exemplify this classical approach to computational OT.

(1) Computational problems that arise in modeling the acquisition of phonology within the framework of OT are tackled by means of *ad hoc* combinatorial algorithms, specifically tailored to the exotic framework of OT, developed from scratch with no connections to methods and results in Machine Learning.

In order to bridge this gap between Computational Phonology and Machine Learning, various scholars have recently started to entertain and explore variants of OT that replace constraint ranking with *constraint weighting* and strict domination with *additive interaction*, and thus fall within the general class of *linear models* very well studied in Machine Learning. An important and simple such model is *Harmonic Grammar* (henceforth: HG); see Legendre et al. (1990b,a). In section 2, I birefly review the two frameworks of OT and HG, in order to introduce the notation used throughout the paper. Claim (2) has thus become rather common in the recent Computational Phonology literature; see for instance Potts et al. (2010), Pater (2009), Hayes and Wilson (2008), Coetzee and Pater (2008), Boersma and Pater (2007, 2008), Jesney and Tessier (2007, 2008), among others. Here are

[1] A framework close in spirit to OT was popular in the Operation Research literature in the Seventies; see for instance Fishburn (1974) for a review. As a metter of fact, Tesar and Smolensky's (1998) Recursive Constraint Demotion was later re-descovered within the Operation Research literature; see Dombi et al. (2007).

some quotes from Pater (2009) that exemplify claim (2): "[I will] illustrate and extend existing arguments for the replacement of OT's ranked constraints with [HG's] weighted ones: that the resulting model of grammar [...] is compatible with well-understood algorithms for learning and other computations. [...] The strengths of HG in this area are of considerable importance" (p. 1002); "This section briefly reviews and expands upon published arguments for the replacement of OT ranking with HG weighting. [...] One broad argument for weighted constraints [...] is that weighed constraints are compatible with existing well-understood algorithms for learning variable outcomes and for learning gradually [...]. As these algorithms are broadly applied with connectionist and statistical models of cognition, this forms an important connection between the HG version of Generative Linguistics and other research in cognitive science" (p. 1021).

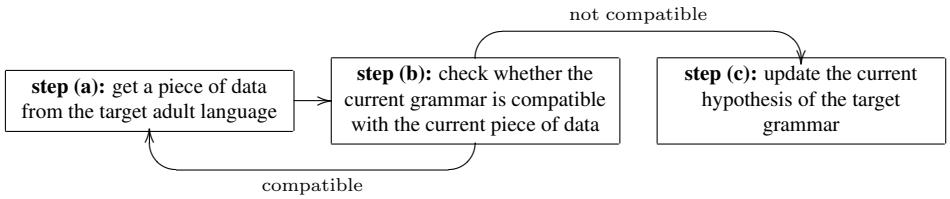
(2) HG is computationally superior to OT because it comes with well established algorithms from Machine Learning (i.e. algorithms for linear classification), contrary to OT.

In section 3, I *prove* that claim (2) is false. In fact, I show a simple trick that allows algorithms for HG to be extended to OT. Thus, HG has no computational advantages over OT and the departure from OT to HG is not warranted on the basis of computational considerations.[2] This result is important because it opens the way to the new approach to computational OT described in (3), radically different from the approach (1) pursued so far in the literature.

(3) Computational problems within the framework of OT can be tackled by adapting well known algorithms from Machine Learning, rather than by devising from scratch *ad hoc* combinatorial algorithms.

In section 4, I illustrate the fruitfulness of this new approach (3) to Computational OT by showing that it leads to substantial progress in the theory of OT online learning algorithms. An online learning algorithm maintains a current hypothesis of the target grammar and updates this hypothesis by repeating the three steps in (4): first, the algorithm receives a piece of data; then, the algorithm checks whether its current hypothesis is compatible with that current piece of data; if it is not, then the algorithm takes action, by updating its current hypothesis to a slightly modified hypothesis. Online algorithms are interesting both from a modeling and a computational perspective. In fact, they define a sequence of grammatical hypotheses that can be matched with attested acquisition paths. Furthermore, online algorithms are memoryless, namely they do not keep track of previously seen forms, and thus do not impose unrealistic memory requirements. Because of their cognitive plausibility, online algorithms are the main modeling tool of the OT acquisitional literature; see Gnanadesikan (2004), Levelt et al. (2000) and Bernhardt and Stemberger (1998) for classical examples. Online algorithms also raise very interesting computational challenges, as the globally correct final hypothesis needs to arise as the consequence of small instantaneous choices based on a single data point at the time. They have thus been carefully studied in the Machine Learning literature; see for instance Cesa-Bianchi and Lugosi (2006) for a review.

(4)



One of the main open issues in Computational OT is how to adapt the general online scheme (4) to the case of OT. The most influential proposal is due to Boersma (1997, 1998). His algorithm maintains a numerical representation of (the ranking corresponding to) the current OT grammar. In step (4b), the algorithm checks whether the current OT-grammar is consistent with the current piece of data. If it isn't, then the algorithm takes action, by updating the (numerical representation of the) current ranking as in (5): virtuous (offending) constraints are promoted (demoted) by a small amount, say

[2] Of course, this claim does not in any way provide an argument *in favor* of OT and against alternative frameworks such as HG. It only shows that the argument (2) against OT and in favor of HG does not go through.

1. The OT online algorithm thus obtained is called the (deterministic)[3] *Gradual Learning Algorithm* (henceforth: GLA).

(5) a. Promote virtuous (i.e. winner-preferring) constraints by 1;
b. demote offending (i.e. loser-preferring) constraints by 1.

Various studies have shown the good modeling capabilities of this simple algorithm; see for instance Boersma and Levelt (2000), Curtin and Zuraw (2002), Boersma and Hayes (2001), etcetera. Yet, the algorithm has resisted theoretical analysis and in particular its convergence has remained an open issue for many years; see Keller and Asudeh (2002) for discussion. Until the issue has been recently settled by Pater (2008), who has shown that the GLA does not converge via a simple counterexample. One of the main open questions in Computational OT is thus the following: is it possible to devise a variant of the GLA that provably converges, so as to retain its modeling virtues without sacrificing computational soundness? In section 4, I show that the non-classical approach to Computational OT sketched in (3) leads to a simple solution to this important question. I argue that the intrinsic ranking logic of OT suggests that the promotion amount of 1 in (5a) should actually be split over the various virtuous constraints, so that each of them should be promoted only by the inverse of the total number of virtuous constraints, as in (6a). I prove that the corresponding OT online algorithm converges, building on convergence of the classical *Perceptron* algorithm for HG. Furthermore, I show how to derive bounds on the worst-case number of updates. Finally, I point out that the approach can be generalized from the Perceptron to any other online algorithm for HG, leading to a large number of brand new online algorithms for OT. These computational developments greatly enrich our algorithmic tools for modeling the acquisition of phonology within the mainstream phonological framework of OT.

(6) a. Promote virtuous (i.e. winner-preferring) constraints by the inverse of the total number of virtuous constraints;
b. demote offending (i.e. loser-preferring) constraints by 1.

Somewhat surprisingly, these powerful algorithmic applications can be developed at an extremely elementary level. Thus, the paper should be accesible also to the reader with no computational inclination. Detailed proofs are collected in the final Appendix.

2 Description of the frameworks of OT and HG

Subsection 2.1 introduces the two frameworks of OT and HG, highlighting the deep commonalities between the two frameworks. Subsection 2.2 introduces the "comparative notation" for the two frameworks, namely a compact way of representing data that will turn out particularly useful in the rest of the paper. Subsection 2.3 reviews what is currently known concerning the relationship between the two frameworks.

2.1 Basic description of HG and OT

Both HG and OT typologies are defined on the background of a 4-tuple $(\mathcal{X}, \mathcal{Y}, Gen, \mathcal{C})$ of *typological specifications*, as in (7a). The first two ingredients are the set of *underlying forms* $\mathcal{X}$ and the set of *surface forms* $\mathcal{Y}$. The third ingredient is a *generating function* Gen that maps an underlying form $x \in \mathcal{X}$ into a set $Gen(x) \subseteq \mathcal{Y}$ of surface forms, called the *candidates* for that underlying form. The fourth ingredient is a *constraint set* $\mathcal{C}$ that contains n functions $C_1, \ldots, C_n$ called *constraints*. Each constraint C_k maps a pair (x, y) of an underlying form $x \in \mathcal{X}$ and a corresponding candidate form $y \in Gen(x)$ into a (nonnegative) integer $C_k(x, y)$, called the *number of violations* assigned by constraint C_k to the mapping of the underlying form x to the candidate surface form y. An example of typological specifications is provided in (7b): the set of underlying forms $\mathcal{X}$ and the set of

[3] Boersma also considers a stochastic variant of the GLA, whereby a small additive gaussian error is added to the (numerical representation of the) current ranking. In this paper, I focus on deterministic OT online algorithms, and thus defer to future research how the results presented here might extend to the stochastic GLA.

surface forms $\mathcal{Y}$ coincide; the generating function Gen is only allowed to modify voicing, but does not perform neither deletion nor epenthesis; the constraint set $\mathcal{C}$ contains a markedness constraint against voiced obstruents and two variants of the faithfulness constraint for voicing, a general and a positional one.

(7) a. $\mathcal{X}$ = set of *underlying forms*;
$\mathcal{Y}$ = set of *surface forms*;
Gen = generating function
$\mathcal{C}$ = constraint set

b. $\mathcal{X} = \{$/ta/, /da/, /rat/, /rad/$\}$
$\mathcal{Y} = \{$[ta], [da], [rat], [rad]$\}$
$Gen = \begin{bmatrix} \text{/ta/, /da/} \rightarrow \{\text{[ta], [da]}\}, \\ \text{/rad/, /rat/} \rightarrow \{\text{[rat], [rad]}\} \end{bmatrix}$
$\mathcal{C} = \left\{ \begin{array}{l} F_{\text{pos}} = \textsc{Ident}[\textsc{voice}]/\textsc{onset}, \\ F_{\text{gen}} = \textsc{Ident}[\textsc{voice}], \\ M = {}^*[+\textsc{voice}, -\textsc{sonorant}] \end{array} \right\}$

The basic data unit in both HG and OT is a *data triplet* as in (8a), consisting of an underlying form $x \in \mathcal{X}$ and two corresponding candidates $\hat{y}, y \in Gen(x)$, with the understanding that the first candidate $\hat{y}$ is the intended *winner* surface form corresponding to the underlying form x while the other candidate y is an intended *loser* surface form. An example of an underlying/winner/loser form data triplet is provided in (8b): the underlying form /rad/ is paired up with the two candidate surface forms [rat] and [rad], together with the information that the former is the intended winner while the latter is a loser.

(8) a. $(\ x,\ \underset{\text{winner}}{\hat{y}},\ \underset{\text{loser}}{y}\ )$

b. (/rad/, [rat], [rad])
winner: [rat]; loser: [rad]

An HG grammar is parameterized by a *weight vector*, which is a tuple $\boldsymbol{\theta}$ with n numerical components $\theta_1, \ldots, \theta_n$ (one for every constraint), as in (9). The kth component θ_k is called the *weight* of the corresponding constraint C_k. An example is provided in (9b) for the case of the constraint set in (7b): this constraint set contains three constraints, namely F_{pos}, F_{gen} and M, labeled C_1, C_2 and C_3; the weight vector $\boldsymbol{\theta}$ assigns them the three weights 8, 2, and 4, respectively.

(9) a. $\boldsymbol{\theta} = \big(\ \overset{C_1}{\theta_1},\ \ldots\ \overset{C_k}{\underset{\text{weight of } C_k}{\theta_k}},\ \ldots\ \overset{C_n}{\theta_n}\ \big)$

b. $\boldsymbol{\theta} = \big(\ \overset{C_1=F_{\text{pos}}}{8}\quad \overset{C_2=F_{\text{gen}}}{2}\quad \overset{C_3=M}{4}\ \big)$

To complete the description of the HG framework, we need a notion of "compatibility" between a hypothesis (i.e. a weight vector) and a piece of data (i.e. a data triplet). A weight vector $\boldsymbol{\theta}$ is called *HG-compatible* with an underlying/winner/loser form data triplet $(x,\ \hat{y},\ y)$ iff condition (10) holds. Condition (10) says that the intended loser y violates the constraints "more severely" than the intended winner $\hat{y}$. In the sense that the sum of the constraint violations for the loser y multiplied by the corresponding weights is (strictly) larger than the sum of the constraint violations for the winner $\hat{y}$ multiplied by the corresponding weights. Of course, a weight vector $\boldsymbol{\theta}$ is called HG-compatible with a set of data triplets iff it is HG-compatible with every triplet in the set. And a set of data triplets is called HG-compatible iff it is compatible with at least a weight vector. As an example, note that the weight vector (9b) is HG-compatible with the data triplet (8b).

(10) $$\sum_{k=1}^{n} \theta_k \cdot \underset{\substack{\text{violations of} \\ \text{the loser } y}}{C_k(x, y)} > \sum_{k=1}^{n} \theta_k \cdot \underset{\substack{\text{violations of} \\ \text{the winner } \hat{y}}}{C_k(x, \hat{y})}$$

If the weights are allowed to be negative, unwanted typological consequences follow. Here is an example. Consider again the typological specifications in (7b). If we allow for negative weights,

then the triplet (/ta/, [da], [ta]) turns out to be HG-compatible. This means that [da] wins over [ta] as the surface form corresponding to the underlying form /ta/. This result is undesired, as the underlying form /ta/ is unmarked and should therefore always surface faithfully. Thus, from now on I will restrict the definition of weight vectors by requiring the weights to be nonnegative, as stated in (11). Let me stress that this non-negativity restriction (11) is not part of the core computational definition of the model, and it can thus be relaxed if needed. Rather, condition (11) is due to the fact that constraints are only interpreted as assigning "violations", rather than "rewards".

(11) $\theta_1, \ldots, \theta_n \geq 0$

In this paper, I will be concerned with the classical computational problem (12a), that I will refer to as the *Weighting problem*. This is the simplest computational problem in HG. On the one hand, this problem is simple because the input to the problem is as rich as possible: the underlying forms are provided as well as the losers, and the data are guaranteed to be HG-compatible. On the other hand, this problem is simple because the output of the problem is as unconstrained as possible: the output weight vector is only required to be HG-compatible with the data, without any further requirement. I will denote by $\mathrm{WP}(\mathcal{D})$ the instance of the Weighting problem (12a) corresponding to a set of data triplets $\mathcal{D}$, or equivalently the set of all its solutions. I will denote by $\mathrm{WP}_{\mathrm{unr}}(\mathcal{D})$ the variant of problem (12a) without the non-negativity restriction (11) on the weight vectors. An example of Weighting problem is provided in (12b): we are given the underlying forms /da/ and /rad/ together with the corresponding intended winner surface forms [da] and [rat] and the corresponding intended losers [ta] and [rad], respectively; we are asked to come up with (nonnegative) weights for the constraints in (7b) that are HG-compatible with these data. One solution of this instance (12b) of the Weighting problem is the weight vector in (9b).

(12) a. *given:* a finite HG-compatible data set $\mathcal{D}$ of underlying/winner/loser form triplets;
find: a non-negative weight vector $\boldsymbol{\theta}$ HG-compatible with the set of data triplets $\mathcal{D}$, according to condition (10).

b. *given:* the two data triplets (/da/, [da], [ta]) and (/rad/, [rat], [rad]);
find: a non-negative weight vector $\boldsymbol{\theta}$ for the constraint set in (7b) HG-compatible with the two data triplets, according to condition (10).

Let me now turn to OT. An OT-grammar is parameterized by a *ranking*, which is a linear order $\gg$ over the constraint set $\mathcal{C}$, as illustrated in (13a), or equivalently in (13a′). A constraint C_h is $\gg$-*ranked above* another constraint C_k iff $C_h \gg C_k$. An example of ranking over the constraint set in (7b) is provided in (13b): it sandwiches the markedness constraint in between the two faithfulness constraints, with the positional faithfulness constraint ranked at the top.

(13) a. top ranked
$C_1 \gg C_2 \gg \ldots \gg C_n$
bottom ranked

a′. C_1
C_2
$\vdots$
C_n

b. F_{pos}
M
F_{gen}

Also in the case of OT, data units are underlying/winner/loser form triplets, as in (8). To complete the definition of the OT framework, we need a notion of "compatibility" between a hypothesis (i.e. a ranking) and a piece of data (i.e. a data triplet). A ranking $\gg$ is called *OT-compatible* with an underlying/winner/loser form data triplet $(x, \hat{y}, y)$ iff condition (14) holds. Condition (14) says that the intended loser y violates the constraints "more severely" than the intended winner $\hat{y}$. In the sense that, among those constraints that distinguish between the winner $\hat{y}$ and the loser y, the top ranked one assigns more violations to the loser than to the winner. Of course, a ranking $\gg$ is called OT-compatible with a set of data triplets iff it is OT-compatible with every triplet in the set. Furthermore, a set of data triplets is called OT-compatible iff it is compatible with at least a ranking. As an example, the ranking $\gg$ in (13b) is OT-compatible with the underlyng/winner/loser form triplet (/rad/, [rat], [rad]) in (8b).

(14) $C_k(x, y) > C_k(x, \hat{y})$ (violations of the loser y; violations of the winner $\hat{y}$)

where C_k = the top $\gg$-ranked constraint among those constraints that assign a different number of violations to the loser y and to the winner $\hat{y}$.

In this paper, I will be concerned with the classical computational problem (15a), that I will refer to as the *Ranking problem*. This is the simplest computational problem in OT. On the one hand, this problem is simple because the input to the problem is as rich as possible: the underlying forms are provided as well as the intended losers; and the data are guaranteed to be OT-compatible. On the other hand, this problem is simple because the output of the problem is as unconstrained as possible: the output ranking is only required to be OT-compatible with the data, without any further requirement. I will denote by RP($\mathcal{D}$) the instance of the Ranking problem (15a) corresponding to a set of data triplets $\mathcal{D}$, or equivalently the set of its solutions. An example of Ranking problem is provided in (15b): we are provided with the underlying forms /da/ and /rad/ together with the corresponding intended winner surface forms [da] and [rat] and the corresponding intended losers [ta] and [rad], respectively; we are asked to come up with a ranking of the constraint set in (7b) OT-compatible with these data. Of course, the unique solution of this instance (15b) of the Ranking problem is the ranking in (13b).

(15) a. *given:* a finite OT-compatible data set $\mathcal{D}$ of underlying/winner/loser form triplets;
find: a ranking $\gg$ OT-compatible with the set of data triplets $\mathcal{D}$, according to condition (14).

b. *given:* the two data triplets (/da/, [da], [ta]) and (/rad/, [rat], [rad]);
find: a ranking $\gg$ of the constraint set in (7b) OT-compatible with the two data triplets, according to condition (14).

This concludes the basic description of the two frameworks of HG and OT, as needed for the rest of the paper. In the next subsection, I will introduce a more compact notation for the data in the two frameworks. This more compact notation will turn out very useful in the rest of the paper.

2.2 A more compact notation for the data in HG and OT

Given an underlying/winner/loser form data triplet $(x,\ \hat{y},\ y)$, the quantity (16a) is called the *constraint difference* on that data triplet for constraint C_k. It is the difference between the number $C_k(x,\ y)$ of violations w.r.t. constraint C_k incurred by the intended loser y and the number $C_k(x,\ \hat{y})$ of violations w.r.t. that same constraint C_k incurred by the intended winner $\hat{y}$. An example is provided in (16b) with respect to the underlying/winner/loser form data triplet in (8b) and the constraint set in (7b). The constraint difference corresponding to F_{pos} is zero, because that constraint assigns the same number of violations (namely 0) to the mapping of /rad/ to [rat] and to [rad]. The constraint difference corresponding to F_{gen} is -1, because the intended loser [rad] is fully faithful to the underlying form /rad/ contrary to the intended winner [rat]. Finally, the constraint difference corresponding to M is 1, as only the intended loser [rad] violates the markedness constraint, not the intended winner [rat].

(16) a. $C_k(x, y) - C_k(x, \hat{y})$ (violations of the loser y; violations of the winner $\hat{y}$)

b.
$$\begin{aligned} F_{\text{pos}}\big(\text{/rad/},\ [\text{rad}]\big) - F_{\text{pos}}\big(\text{/rad/},\ [\text{rat}]\big) &= 0 \\ F_{\text{gen}}\big(\text{/rad/},\ [\text{rad}]\big) - F_{\text{gen}}\big(\text{/rad/},\ [\text{rat}]\big) &= -1 \\ M\big(\text{/rad/},\ [\text{rad}]\big) - M\big(\text{/rad/},\ [\text{rat}]\big) &= 1 \end{aligned}$$
(loser mapping; winner mapping)

Condition (10) for HG-compatibility can of course be rewritten as in (17), by bringing everything on one side. This restatement highlights the fact that HG-compatibility is only sensitive to the constraint differences, not to the actual numbers of constraint violations.

$$(17)\quad \sum_{k=1}^{n} \theta_k \cdot \underbrace{\Big(C_k(x,y) - C_k(x,\hat{y})\Big)}_{k\text{th constraint difference}} > 0$$

Thus, the information provided by a data triplet that is useful for the sake of establishing HG-compatibility can be distilled as in (18). The data triplet is paired up with a tuple with n entries (one for every constraint), with the convention that the kth entry is the kth constraint difference $C_k(x,y) - C_k(x,\hat{y})$. One such n-tuple of numbers is called an *HG-comparative row*, since it contains all the information that is needed to compare the winner and the loser within HG. I denote HG-comparative rows by $\overline{\mathbf{a}}$ and their components by $\overline{a}_1, \ldots, \overline{a}_n$. An example is provided in (18b) for the case of the data triplet (8b), building on the computation (16b).

(18) a. $(x, \hat{y}, y) \implies \overline{\mathbf{a}} = \begin{bmatrix} \overline{a}_1 & \ldots & \overline{a}_n \end{bmatrix}$ where $\overline{a}_k = C_k(x,y) - C_k(x,\hat{y})$ (winner: y; loser: $\hat{y}$)

b. (/rad/, [rat], [rad]) $\implies$ $\begin{bmatrix} F_{\text{pos}} & F_{\text{gen}} & M \\ 0 & -1 & 1 \end{bmatrix}$ (winner: [rat]; loser: [rad])

If we have many, say m, data triplets, then we can pair up each of them with the corresponding HG-comparative row as in (18) and we can organize these m HG-comparative rows one underneath the other (the order does not matter), into an *HG-comparative tableau* with n columns (one for every constraint), m rows (one for every triplet) and numerical entries corresponding to constraint differences, as in (19a). I denote by $\overline{\mathbf{A}}(\mathcal{D})$ the HG-comparative tableau corresponding to a set of data triplets $\mathcal{D}$; I denote by $\overline{\mathbf{A}}$ an arbitrary HG-comparative tableau; I often omit zeros for readability. An example is provided in (19b): it has three columns, because the constraint set in (7b) contains three constraints; it has two rows, because it corresponds to the two data triplets (/da/, [da], [ta]) and (/rad/, [rat], [rad]); its entries are numbers according to (18).

(19) a.
$$\overline{\mathbf{A}} = \underbrace{\left.\begin{bmatrix} 5 & -2 & 1 & -1 & 0 \\ -1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & -2 & -3 \end{bmatrix}\right\}}_{n \text{ columns}} m \text{ rows}$$
(c_1 … c_k … c_n)

b.
$$\begin{array}{c} \\ (/\text{da}/, [\text{da}], [\text{ta}]) \\ (/\text{rad}/, [\text{rat}], [\text{rad}]) \end{array} \begin{array}{c} \text{winner} \\ \downarrow \\ \begin{bmatrix} 1 & 0 & -1 \\ 0 & -1 & 1 \end{bmatrix} \\ \text{loser} \end{array} \quad \begin{matrix} F_{\text{pos}} & F_{\text{gen}} & M \end{matrix}$$

With this notation in place, condition (17) for HG-compatibility between a weight vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ and an underlying/winner/loser form data triplet can be restated in terms of the corresponding HG-comparative row $\overline{\mathbf{a}} = (\overline{a}_1, \ldots, \overline{a}_n)$ as condition (20). Thus, let's say that a weight vector $\boldsymbol{\theta}$ is *HG-compatible* with an arbitrary HG-comparative row $\overline{\mathbf{a}} = (\overline{a}_1, \ldots, \overline{a}_n)$ iff condition (20) holds. Of course, a weight vector $\boldsymbol{\theta}$ is called HG-compatible with an arbitrary HG-comparative tableau $\overline{\mathbf{A}}$ iff it is HG-compatible with every row of the tableau. And an HG-comparative tableau is called HG-compatible iff it is compatible with at least a weight vector.

$$(20)\quad \sum_{k=1}^{n} \theta_k \overline{a}_k > 0.$$

The Weighting problem has been stated in (12a) in terms of data triplets. As noted above, actual data triplets carry superfluous information. And a sharper representation of data triplets is provided by HG-comparative rows and tableaux. Thus, it is convenient to restate the Weighting problem in terms of HG-comparative tableaux, as in (21a). I will denote by $\text{WP}(\overline{\mathbf{A}})$ the instance of the Weighting problem (21) corresponding to an HG-comparative tableau $\overline{\mathbf{A}}$, or equivalently the set of its solutions. Of course, a weight vector is a solution of the instance of the original Weighting problem (12a) for a

given set of data triplets iff it is a solution of the instance of the problem (21a) for the corresponding HG-comparative tableau, namely $\mathrm{WP}(\mathcal{D}) = \mathrm{WP}(\overline{\mathbf{A}}(\mathcal{D}))$. I will denote by $\mathrm{WP}_{\mathrm{unr}}(\overline{\mathbf{A}})$ the variant of problem (21a) without the non-negativity restriction (11) on the weight vectors. As an example, I give in (21b) the restatement of the Weighting problem (12b) in comparative notation.

(21) a. *given:* an HG-compatible HG-comparative tableau $\overline{\mathbf{A}}$;
find: a non-negative weight vector $\boldsymbol{\theta}$ HG-compatible with the tableau $\overline{\mathbf{A}}$, according to condition (20).

b. *given:* the HG-comparative tableau $\overline{\mathbf{A}}$ in (19b);
find: a non-negative weight vector $\boldsymbol{\theta}$ for the constraint set in (7b) HG-compatible with $\overline{\mathbf{A}}$, according to condition (20).

An analogous simplification of the representation of the data and of the corresponding core computational problem is available within the framework of OT. Given an underlying/winner/loser form data triplet $(x, \hat{y}, y)$, the constraints can be sorted into *winner-preferring*, *loser-preferring* or *even* as in (22a), depending on whether the corresponding constraint difference is positive (i.e. the constraint assigns more violations to the loser than to the winner), negative (i.e. the constraint assigns less violations to the loser than to the winner) or null (i.e. the constraint assigns the same number of violations to the loser and to the winner). An example is provided in (22b) for the constraint set in (7b) and the data triplet in (8b), building on the computations in (16b).

(22) a.

$$\text{Constraint } C_k \text{ is } \left\{ \begin{array}{c} \textit{winner-preferring} \\ \textit{loser-preferring} \\ \textit{even} \end{array} \right\} \text{ iff } \left\{ \begin{array}{l} C_k(x,y) - C_k(x,\hat{y}) > 0 \\ C_k(x,y) - C_k(x,\hat{y}) < 0 \\ C_k(x,y) - C_k(x,\hat{y}) = 0 \end{array} \right\}$$

($\hat{y}$: violations of the winner; y: violations of the loser)

b.

(/rad/, [rad], [rat])	$\Longrightarrow$		
(winner / loser)		C_1 = IDENT[VOICE]/ONSET:	even
		C_2 = IDENT[VOICE]:	winner-preferring
		C_3 = *[VOICE]:	loser-preferring

The notion of OT-compatibility in (14) only depends on whether the various constraints are winner-preferrer, loser-preferrer or even. Following Tesar (1995) and Prince (2002), the information provided by a data triplet that is useful for the sake of OT-compatibility can thus be distilled as in (23a). The data triplet is paired up with a tuple with n entries (one for every constraint), with the convention that the kth entry is equal to W, L or E depending on whether the kth constraint C_k is winner-preferrer or loser-preferrer or even. One such n-tuple of L's, E's and W's is called an *OT-comparative row*, since it contains all the information that is needed to compare the winner and the loser within OT.[4] I denote an OT- comparative row by $\mathbf{a}$ and its entries by $a_1, \ldots, a_n$. An example is provided in (23b) for the case of the data triplet in (8b), based on the classification of the constraint set (7b) already computed in (22b).

(23) a. $(x, \hat{y}, y) \Longrightarrow \mathbf{a} = \begin{bmatrix} a_1 & \ldots & a_n \end{bmatrix}$ (winner: $\hat{y}$; loser: y)

$$a_k = \left\{ \begin{array}{ll} \text{W} & \text{if } C_k \text{ is winner-preferrer} \\ \text{L} & \text{if } C_k \text{ is loser-preferrer} \\ \text{E} & \text{if } C_k \text{ is even} \end{array} \right.$$

[4] Various alternative names for OT-comparative rows have been used in the literature: Prince (2002) calls them *elementary ranking conditions*; Tesar and Smolensky (1998) call them *mark data pairs*.

b.
$$\begin{array}{c} \text{winner} \\ | \\ (/\text{rad}/,\ [\text{rat}],\ [\text{rad}]) \\ | \\ \text{loser} \end{array} \quad \Longrightarrow \quad \begin{array}{c} C_1{=}F_{\text{pos}} \quad C_2{=}F_{\text{gen}} \quad C_3{=}M \\ \left[\begin{array}{ccc} \text{E} & \text{L} & \text{W} \end{array}\right] \end{array}$$

If we have many, say m, data triplets, then we can pair up each of them with the corresponding OT-comparative row as in (23a) and we can organize these m OT-comparative rows one underneath the other (the order does not matter), into an *OT-comparative tableau* with n columns (one for every constraint), m rows (one for every data triplet) and entries equal to W, L and E, as in (24a). I denote by $\mathbf{A}(\mathcal{D})$ the OT-comparative tableau corresponding to a set of data triplets $\mathcal{D}$; I denote by $\mathbf{A}$ an arbitrary OT-comparative tableau; I often omit E's for readability. An example is provided in (24b): it has three columns, because the constraint set in (7b) contains three constraints; it has two rows, because it corresponds to the two data triplets (/da/, [da], [ta]) and (/rad/, [rat], [rad]); its entries are W's, L's and E's according to rule (23).

(24) a.
$$\mathbf{A} = \underbrace{\left.\begin{array}{ccccc} C_1 & \dots & C_k & \dots & C_n \\ \left[\begin{array}{ccccc} \text{W} & \text{L} & \text{W} & \text{L} & \text{E} \\ \text{L} & \text{W} & \text{W} & \text{E} & \text{E} \\ \text{E} & \text{W} & \text{W} & \text{L} & \text{L} \end{array}\right] \end{array}\right\} m \text{ rows}}_{n \text{ columns}}$$

b.
$$\begin{array}{c} \text{winner} \\ | \\ (/\text{da}/,\ [\text{da}],\ [\text{ta}]) \\ (/\text{rad}/,\ [\text{rat}],\ [\text{rad}]) \\ | \\ \text{loser} \end{array} \quad \begin{array}{c} F_{\text{pos}} \quad F_{\text{gen}} \quad M \\ \left[\begin{array}{ccc} \text{W} & \text{W} & \text{L} \\ \text{E} & \text{L} & \text{W} \end{array}\right] \end{array}$$

With this notation in place, condition (14) for OT compatibility between a ranking $\gg$ and a set of underlying/winner/loser form data triplets can be restated as condition (25a) in terms of the corresponding OT-comparative tableau. Thus, let's say that a ranking $\gg$ is *OT-compatible* with an arbitrary comparative tableau iff condition (25a) holds. Of course, an OT-comparative tableau is called OT-compatible iff it is compatible with at least a ranking. To illustrate the notion of OT-compatibility in (25a), note that the OT-comparative tableau in (24b) is OT-compatible with the ranking $F_{\text{pos}} \gg M \gg F_{\text{gen}}$ in (13b): once its columns are ordered from left to right in $\gg$-decreasing order, we obtain the tableau (25b), that has indeed the property that the leftmost non-E symbol of every row is a W.

(25) a. Once the n entries of the tableau are reordered from left to right in decreasing order according to $\gg$, then the leftmost non-E entry is a W.

b.
$$\begin{array}{c} \curvearrowright \\ F_{\text{pos}} \quad M \quad F_{\text{gen}} \\ \left[\begin{array}{ccc} \text{W} & \text{L} & \text{W} \\ \text{E} & \text{W} & \text{L} \end{array}\right] \end{array}$$

The Ranking problem has been stated in (15a) in terms of data triplets. As noted above, actual data triplets carry superfluous information. And a sharper representation of data triplets is provided by OT-comparative rows and tableaux. Thus, it is convenient to restate the Ranking problem in terms of OT-comparative tableaux, as in (26a). I will denote by $\text{RP}(\mathbf{A})$ the instance of the Ranking problem (26a) corresponding to an OT-comparative tableau $\mathbf{A}$, or equivalently the set of its solutions. Of course, a ranking is a solution of the instance of the original Ranking problem (15a) for a given set of data triplets iff it is a solution of the instance of the problem (26a) for the corresponding comparative tableau, namely $\text{RP}(\mathcal{D}) = \text{RP}(\mathbf{A}(\mathcal{D}))$. As an example, I give in (26b) the formulation of the Ranking problem (15b) in comparative notation.

(26) a. *given:* an OT-compatible OT-comparative tableau $\mathbf{A}$;
find: a ranking $\gg$ OT-compatible with the tableau $\mathbf{A}$, according to condition (25).

b. *given:* the OT-comparative tableau $\mathbf{A}$ in (24b);
find: a ranking $\gg$ of the constraint set in (7b) OT-compatible with $\mathbf{A}$, according to condition (25).

This subsection just restated the two frameworks of HG and OT in terms of the sharper, more compact comparative notation. This restatement was laborious, but it will prove remarkably useful in the rest of the paper.

2.3 What is currently known on the relationship between HG and OT

The following claim 1 summarizes what is currently known in the literature concerning the relationship between the two frameworks of OT and HG; see Prince and Smolensky (2004) and Keller (2000, 2005). For completeness, the proof of this claim is recalled in Appendix A.1. The idea of the proof is that the highest-takes-all behavior of the notion of OT-compatibility (14) can be mimicked by the weighted notion of HG-compatibility (10) as long as we use exponentially spaced weights.

Claim 1 *If a set $\mathcal{D}$ of underlying/winner/loser form triplets is OT-compatible, then it is also HG-compatible. More precisely, let $\gg$ be a ranking OT-compatible with $\mathcal{D}$. Without loss of generality, assume that it is (27a), with C_n ranked at the top, C_{n-1} below it and so on, until the bottom ranked C_1. Then, the weight vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ defined in (27b) is HG-compatible with $\mathcal{D}$*

(27) *a.* C_n | C_{n-1} | $\vdots$ | C_1 *b.* $\theta_n = (\delta+1)^n$, $\theta_{n-1} = (\delta+1)^{n-1}$, $\vdots$, $\theta_1 = (\delta+1)$

where δ is the largest constraint difference (ignoring sign) over all constraints and all data triplets in the data set $\mathcal{D}$. ■

Let me illustrate claim 1 with an example. Given the typological specifications in (7b), consider the data set $\mathcal{D}$ consisting of two underlying/winner/form triplets (/da/, [da], [ta]) and (/rad/, [rat], [rad]). The corresponding HG-comparative tableaux of constraint differences is (19b) and the corresponding OT-comparative tableuax is (24b), both repeated in (28). The OT-comparative tableau $\mathbf{A}$ is OT-compatible with the ranking $F_{\text{pos}} \gg M \gg F_{\text{gen}}$ in (13b). Since in this case $\delta = 1$, the corresponding weight vector according to (27) is $\boldsymbol{\theta} = (\theta_{F_{\text{pos}}}, \theta_{F_{\text{gen}}}, \theta_M) = (8, 2, 4)$ in (9b). The latter is indeed HG-compatible with the HG-comparative tableau $\overline{\mathbf{A}}$.

(28) a. $\overline{\mathbf{A}} = \begin{array}{c} \begin{array}{ccc} F_{\text{pos}} & F_{\text{gen}} & M \end{array} \\ \left[\begin{array}{ccc} 1 & 0 & -1 \\ 0 & -1 & 1 \end{array}\right] \end{array}$ b. $\mathbf{A} = \begin{array}{c} \begin{array}{ccc} F_{\text{pos}} & F_{\text{gen}} & M \end{array} \\ \left[\begin{array}{ccc} \text{W} & \text{W} & \text{L} \\ \text{E} & \text{L} & \text{W} \end{array}\right] \end{array}$

The reverse of claim 1 does not hold, namely there exist data sets $\mathcal{D}$ that are HG-compatible but not OT-compatible. Here is a counterexample. Suppose that the HG-comparative tableau of constraint differences is $\overline{\mathbf{A}}$ in (29a). The corresponding OT-comparative tableau is $\mathbf{A}$ in (29b). The former is HG-compatible (say with the weights $\theta_1 = 3$ and $\theta_2 = \theta_3 = 2$) but the latter is not OT-compatible.

(29) a. $\overline{\mathbf{A}} = \begin{array}{c} \begin{array}{ccc} C_1 & C_2 & C_3 \end{array} \\ \left[\begin{array}{ccc} 1 & -1 & 0 \\ 1 & 0 & -1 \\ -1 & 1 & 1 \end{array}\right] \end{array}$ b. $\mathbf{A} = \begin{array}{c} \begin{array}{ccc} C_1 & C_2 & C_3 \end{array} \\ \left[\begin{array}{ccc} \text{W} & \text{L} & \text{E} \\ \text{W} & \text{E} & \text{L} \\ \text{L} & \text{W} & \text{W} \end{array}\right] \end{array}$

The point of example (29) can be made more explicit as follows. In order for a weight vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ to be HG-compatible with the first and second rows of the HG-comparative tableau (29a), the weight of constraint C_1 has got to be larger than both the weights of C_2 and C_3, as in (30a); in order for a ranking $\gg$ to be OT-compatible with the first and second row of the OT-comparative tableau (29b), constraint C_1 has got to be ranked above both constraints C_2 and C_3, as in (30b). No ranking that satisfies the ranking conditions (30b) can ever be OT-compatible with the third row of the OT-comparative tableau (29b). A weight vector that satisfies the weighting conditions (30a) can instead be HG-compatible with the third row of the HG-comparative tableau (29a), provided that the two constraints C_2 and C_3, despite their small weight, are allowed to join forces and gang up against C_1, in the sense that the sum $\theta_2 + \theta_3$ of their weights is larger than the weight θ_1 of constraint C_1. The crucial difference between HG and OT is that the former allows for these *gang-up effects*, while the latter doesn't.

(30) a. $\theta_1 > \theta_2$, $\theta_1 > \theta_3$ b. $C_1 \gg C_2$, $C_1 \gg C_3$

Claim 1 can be restated as follows from an algorithmic point of view. Suppose we are given an instance WP($\mathcal{D}$) of the Weighting problem (12) corresponding to a data set $\mathcal{D}$ that happens to be not only HG-compatible but actually also OT-compatible. Consider the Weighting problem WP($\overline{\mathbf{A}}$) corresponding to the HG-comparative tableau $\overline{\mathbf{A}}$ corresponding to the data set $\mathcal{D}$. Claim 1 says that, instead of solving the Weighting problem WP($\overline{\mathbf{A}}$) *directly*, we can solve it *indirectly*, through the three steps (31a)-(31c): first, we construct the corresponding OT-comparative tableau $\mathbf{A}$ out of the tableau of constraint differences $\overline{\mathbf{A}}$, as in (23); then, we solve the corresponding instance RP($\mathbf{A}$) of the Ranking problem (26) instead; finally, we obtain a weight vector that solves the given Weighting problem WP($\overline{\mathbf{A}}$), through (27).

(31)
$$\begin{array}{ccc} \overline{\mathbf{A}} & \xrightarrow{\text{WP}} & \boldsymbol{\theta} \\ {\scriptstyle (a)}\downarrow & & \uparrow{\scriptstyle (c)} \\ \mathbf{A} & \xrightarrow[(b)]{\text{RP}} & \gg \end{array}$$

In conclusion, claim 1 says that the Weighting problem can be reduced to the Ranking problem, at least in certain cases (namely, when the data set is OT-compatible). Yet, as recalled in section 1, we already know how to solve the Weighting problem, since we can draw on the large literature on linear models; see for instance Potts et al. (2010). What we are really looking for is instead good methods to solve the Ranking problem. The fact that we can reduce the Weighting problem to the Ranking problem is of no algorithmic interest. And claim 1 therefore has no interesting algorithmic implications.

3 HG is not computationally superior to OT

In the preceding section, I have introduced the two frameworks of OT and HG and the two corresponding core computational problems, namely the Ranking problem RP($\mathbf{A}$) and the Weighting problem WP($\overline{\mathbf{A}}$), repeated in (32) and (33).

(32) *given:* an OT-compatible OT-comparative tableau $\mathbf{A}$;
find: a ranking $\gg$ that is OT-compatible with the tableau $\mathbf{A}$.

(33) *given:* an HG-compatible HG-comparative tableau $\overline{\mathbf{A}}$;
find: a nonnegative weight vector $\boldsymbol{\theta}$ HG-compatible with the tableau $\overline{\mathbf{A}}$.

The question addressed in this section can *roughly* be stated as follows: given an arbitrary instance of the Ranking problem (32), is it possible to pair it up with an instance of the Weighting problem (33) such that I get a solution to the former by solving the latter instead? This question can be stated more *precisely* as follows: given an instance RP($\mathbf{A}$) of the Ranking problem, is it possible to find one (or, even better, all) of its solutions without solving the problem *directly* but rather *indirectly*, through the scheme in (34)? The latter scheme can be made explicit as follows: first, we pair up the given OT-comparative tableau $\mathbf{A}$ with an HG-comparative tableau $\overline{\mathbf{A}}$, as in (34a); then, we find a solution $\boldsymbol{\theta}$ of the corresponding Weighting problem WP($\overline{\mathbf{A}}$), as in (34b); finally, we pair up that solution $\boldsymbol{\theta}$ with a ranking $\gg$, as in (34c). We hope that the latter ranking actually solves the instance RP($\mathbf{A}$) of the Ranking problem that we started with.

(34)
$$\begin{array}{ccc} \mathbf{A} & \xrightarrow{\text{RP}} & \gg \\ {\scriptstyle (a)}\downarrow & & \uparrow{\scriptstyle (c)} \\ \overline{\mathbf{A}} & \xrightarrow[(b)]{\text{WP}} & \theta \end{array}$$

The scheme in (34) is the inverse of the scheme (31), that summarizes claim 1. Thus, the question considered in this section is whether the algorithmic perspective of the old claim 1 can be inverted, despite the fact that claim 1 itself cannot be inverted.

3.1 Main claim

In order to implement scheme (34), we need to define the two steps (34a) and (34c), namely we need to find proper ways to pair up OT-comparative rows with HG comparative rows and weight vectors with rankings. Let me introduce the core, very simple idea with a couple of examples. Consider first the case of the OT-comparative row $\mathbf{a}$ in (35). Crucially, this comparative row $\mathbf{a}$ contains a unique entry equal to W. Define the corresponding HG-comparative row as $\overline{\mathbf{a}}$ in (35): the E of C_1 is replaced by 0; the W of C_2 is replaced by 1; and the L of C_3 is replaced by -1. A weight vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ is HG-compatible with this derived HG-comparative row $\overline{\mathbf{a}}$ iff $\theta_2 - \theta_3$ is strictly positive. Equivalently, iff the weight θ_2 corresponding to constraint C_2 is strictly larger than the weight θ_3 corresponding to constraint C_3. Consider a ranking that "respects" the ordering implicit in the relative size of these weights. Any such ranking thus ranks C_2 above C_3. The latter ranking condition ensures OT-compatibility with the OT-comparative row $\mathbf{a}$ we started from.

$$\text{(35)} \qquad \mathbf{a} = \begin{bmatrix} C_1 & C_2 & C_3 \\ \text{E} & \text{W} & \text{L} \end{bmatrix} \quad \Longrightarrow \quad \overline{\mathbf{a}} = \begin{bmatrix} C_1 & C_2 & C_3 \\ 0 & 1 & -1 \end{bmatrix}$$

Consider next the case of the OT-comparative row $\mathbf{a}$ in (36). Crucially, this comparative row $\mathbf{a}$ contains two entries equal to W. Define the corresponding HG-comparative row as $\overline{\mathbf{a}}$ in (36): the two W's of C_1 and C_2 are both replaced by 1; and the L of C_3 is replaced by -2, capturing the fact that this row contains two entries equal to W. A weight vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ is HG-compatible with this derived HG-comparative row $\overline{\mathbf{a}}$ iff $\theta_1 + \theta_2 - 2\theta_3$ is strictly positive. Equivalently, iff $(\theta_1 - \theta_3) + (\theta_2 - \theta_3)$ is strictly positive. This implies in particular that either $(\theta_1 - \theta_3)$ is strictly positive or $(\theta_2 - \theta_3)$ is strictly positive (or both). Again, consider a ranking that "respects" the ordering implicit in the relative size of these weights. Any such ranking either ranks constraint C_1 above C_3 or constraint C_2 above C_3 (or both). The latter ranking condition ensures OT-compatibility with the OT-comparative row $\mathbf{a}$ we started from.

$$\text{(36)} \qquad \mathbf{a} = \begin{bmatrix} C_1 & C_2 & C_3 \\ \text{W} & \text{W} & \text{L} \end{bmatrix} \quad \Longrightarrow \quad \overline{\mathbf{a}} = \begin{bmatrix} C_1 & C_2 & C_3 \\ 1 & 1 & -2 \end{bmatrix}$$

Consider again the OT comparative tableau (29b), repeated as $\mathbf{A}$ in (37). Consider the corresponding HG tableau $\overline{\mathbf{A}}$ in (37). Note that the two L's in the first two rows of $\mathbf{A}$ are replaced with a -1 in $\overline{\mathbf{A}}$, as those rows have a unique winner-preferrer; the L in the last row is instead replaced with a -2, because that row has two winner-preferrers. As noted above, the comparative tableau $\mathbf{A}$ is not OT-compatible. It is easy to check that also the derived tableau $\overline{\mathbf{A}}$ is not HG-compatible: in fact, a weight vector $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ HG-compatible with the first two rows needs to satisfy the two inequalities $\theta_1 > \theta_2$ and $\theta_1 > \theta_3$, respectively; adding them together, we get $2\theta_1 > \theta_2 + \theta_3$; the latter inequality says that $\boldsymbol{\theta}$ is not HG-compatible with the third row of $\overline{\mathbf{A}}$.

$$\text{(37)} \qquad \mathbf{A} = \begin{array}{c} \begin{matrix} C_1 & C_2 & C_3 \end{matrix} \\ \begin{bmatrix} \text{W} & \textcircled{\text{L}} & \text{E} \\ \text{W} & \text{E} & \textcircled{\text{L}} \\ \textcircled{\text{L}} & \text{W} & \text{W} \end{bmatrix} \end{array} \quad \Longrightarrow \quad \overline{\mathbf{A}} = \begin{array}{c} \begin{matrix} C_1 & C_2 & C_3 \end{matrix} \\ \begin{bmatrix} 1 & \textcircled{-1} & 0 \\ 1 & 0 & \textcircled{-1} \\ \textcircled{-2} & 1 & 1 \end{bmatrix} \end{array}$$

The reasoning just illustrated with the three examples (35)-(37) holds in the completely general case, as follows. Given an OT-comparative row $\mathbf{a}$, consider the HG-comparative row $\overline{\mathbf{a}}$ derived from $\mathbf{a}$ as in (38): every entry equal to W in $\mathbf{a}$ corresponds to 1 in $\overline{\mathbf{a}}$; every entry equal to E in $\mathbf{a}$ corresponds to 0 in $\overline{\mathbf{a}}$; and every entry equal to L in $\mathbf{a}$ corresponds to $-w$ in $\overline{\mathbf{a}}$, where w is the total number of entries equal to W in the OT-comparative row $\mathbf{a}$. Let me say that an HG-comparative tableau is *derived* from an OT-comparative tableau iff each row of the former is derived from the latter according to (38). The three examples (35)-(37) illustrate this construction.

$$\text{(38)} \qquad \mathbf{a} = \big(a_1, \ldots, a_n\big) \longrightarrow \overline{\mathbf{a}} = \big(\overline{a}_1, \ldots, \overline{a}_n\big) \text{ such that } \overline{a}_k \doteq \begin{cases} -w & \text{if } a_k = \text{L} \\ 0 & \text{if } a_k = \text{E} \\ 1 & \text{if } a_k = \text{W} \end{cases}$$

Let me say that a ranking $\gg$ is *derived* from a weight vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ iff it is compatible with the order implicitly defined by the relative size of the weights, in the sense that condition (39) holds for every pair of constraints C_h and C_k. Let me unpack this condition, by considering two cases in turn. If all the components of the weight vector $\boldsymbol{\theta}$ are pairwise distinct, the vector $\boldsymbol{\theta}$ admits a unique derived ranking, namely the unique ranking that ranks a constraint C_k above a constraint C_h iff the weight θ_k of C_k is larger than the weight θ_h of C_h. If instead the components of the weight vector $\boldsymbol{\theta}$ are *not* all pairwise distinct, then $\boldsymbol{\theta}$ admits multiple derived rankings, because a tie between two weights can be broken differently by different derived rankings.

(39) $\theta_h > \theta_k \implies C_h \gg C_k$

Here is an example. The weight vector $\boldsymbol{\theta}$ in (40a) has pairwise distinct components, and thus it admits a unique derived ranking, namely the ranking that assigns C_1 to the top, C_2 to the bottom, and C_3 in between. In the case of the weight vector $\boldsymbol{\theta}$ in (40b), the weights of C_2 and C_3 tie; thus, this weight vector admits two derived rankings, that break the tie in two different ways.

(40) a. $\boldsymbol{\theta} = \begin{array}{c} \begin{array}{ccc} C_1 & C_2 & C_3 \end{array} \\ \begin{pmatrix} 100 & 10 & 50 \end{pmatrix} \end{array} \longrightarrow C_1 \gg C_3 \gg C_2$

b. $\boldsymbol{\theta} = \begin{array}{c} \begin{array}{ccc} C_1 & C_2 & C_3 \end{array} \\ \begin{pmatrix} 100 & 50 & 50 \end{pmatrix} \end{array} \begin{array}{l} \nearrow \; C_1 \gg C_3 \gg C_2 \\ \searrow \; C_1 \gg C_2 \gg C_3 \end{array}$

The main result of this section is the following claim 2. This claim says that the scheme (34) holds provided that the mapping (34a) from OT-comparative tableaux to HG-comparative tableaux is defined as in (38) and the mapping (34c) from weight vectors to rankings is defined as in (39). In other words, claim 2 says that it is possible to get a solution of a given instance of the Ranking problem (32) without solving it directly, but rather by solving a corresponding instance of the Weighting problem (33).

Claim 2 *Given an OT-comparative tableau* $\mathbf{A}$*, consider the corresponding HG-comparative tableau* $\overline{\mathbf{A}}$ *derived from* $\mathbf{A}$ *as in (38). If* $\overline{\mathbf{A}}$ *is HG-compatible, then* $\mathbf{A}$ *is OT-compatible. More precisely, if a weight vector* $\boldsymbol{\theta}$ *solves the instance WP*$(\overline{\mathbf{A}})$ *of the Weighting problem (33), then any ranking derived from* $\boldsymbol{\theta}$ *according to (39) solves the instance RP*$(\mathbf{A})$ *of the Ranking problem (32).* ■

The proof of claim 2 is presented in Appendix A.2. It is a trivial generalization of the reasoning illustrated above with the three examples (35)-(37).

3.2 Further consequences

Given an OT-comparative tableau $\mathbf{A}$ and the HG-comparative tableau $\overline{\mathbf{A}}$ derived from $\mathbf{A}$ as in (38), consider the corresponding instance RP$(\mathbf{A})$ of the Ranking problem (26) and the corresponding instance WP$(\overline{\mathbf{A}})$ of the Weighting problem (33). The new claim 2 says that I can obtain *some* of the solutions of the Ranking problem RP$(\mathbf{A})$ by finding some of the solutions of the Weighting problem WP$(\overline{\mathbf{A}})$ instead, and then constructing the corresponding derived rankings through (39). Can I find *all* of the solutions of RP$(\mathbf{A})$ this way? That is indeed the case. In fact, consider a ranking $\gg$ that solves the Ranking problem RP$(\mathbf{A})$. Without loss of generality, assume that it is $C_n \gg C_{n-1} \gg \ldots \gg C_1$ (otherwise, just relabel the constraints). Consider the weight vector $\boldsymbol{\theta}$ defined from $\gg$ as in (27), where δ is the largest entry (ignoring sign) of the derived HG-comparative tableau $\overline{\mathbf{A}}$. Note that $\gg$ is derived from $\boldsymbol{\theta}$, namely satisfies condition (39). Furthermore, claim 1 guarantees that $\boldsymbol{\theta}$ solves the Weighting problem WP$(\overline{\mathbf{A}})$. Claims 1 and 2 together thus entail claim 3. The latter claim says that OT does not raise any new computational challenges beyond HG. Hence, the popular claim (2) quoted at the beginning of this paper is wrong.

Claim 3 *Given an OT-comparative tableau* $\mathbf{A}$*, consider the corresponding HG-comparative tableau* $\overline{\mathbf{A}}$ *derived from* $\mathbf{A}$ *as in (38). Then,* $\overline{\mathbf{A}}$ *is HG-compatible iff* $\mathbf{A}$ *is OT-compatible. Furthermore, a ranking solves the instance RP*$(\mathbf{A})$ *of the Ranking problem (26) iff it is derived through (39) from a weight vector that solves the corresponding instance WP*$(\overline{\mathbf{A}})$ *of the Weighting problem (33).* ■

In order to explore a further consequence of the two claims 1 and 2, let me generalize a bit the notion of derived HG-tableaux, as follows. Given an OT-comparative row $\mathbf{a} = (a_1, \ldots, a_n)$, let me say that an HG-comparative row $\overline{\mathbf{a}} = (\overline{a}_1, \ldots, \overline{a}_n)$ is *derived* from the OT-comparative row $\mathbf{a}$ iff it satisfies the condition in (41): every entry equal to W in $\mathbf{a}$ corresponds to a positive entry in $\overline{\mathbf{a}}$; every entry equal to E in $\mathbf{a}$ corresponds to a zero entry in $\overline{\mathbf{a}}$; and every entry equal to L in $\mathbf{a}$ corresponds to a null or negative entry in $\overline{\mathbf{a}}$. Given an OT-comparative tableau $\mathbf{A}$, let me say that an HG-comparative tableau $\overline{\mathbf{A}}$ with the same number of rows and columns is *derived* from $\mathbf{A}$ iff each row of $\overline{\mathbf{A}}$ is derived from the corresponding row of $\mathbf{A}$ according to (41). The notion of derived HG-tableau in (38) is of course a special case of the more general notion in (41).

$$(41) \quad \mathbf{a} = \big[a_1, \ldots, a_n\big] \longrightarrow \overline{\mathbf{a}} = \big[\overline{a}_1, \ldots, \overline{a}_n\big] \text{ such that } \overline{a}_k \begin{cases} > 0 & \text{if } a_k = \text{W} \\ = 0 & \text{if } a_k = \text{E} \\ \leq 0 & \text{if } a_k = \text{L} \end{cases}$$

The old claim 1 can now be restated once more as follows: if an OT-comparative tableau is OT-compatible, then *every* HG-comparative tableau derived from it according to (41) is HG-compatible. Vice versa, assume that every derived HG-comparative tableau is HG-compatible. Then, in particular, the HG-comparative tableau derived according to (38) is HG-compatible. And claim 2 ensures that the OT-comparative tableau we started from is OT-compatible. By putting the two claims 1 and 2 together, we thus obtain claim 4. The latter claim offers a new characterization of OT-compatibility.

Claim 4 *A comparative tableau* $\mathbf{A}$ *is OT-compatible iff* every *HG-comparative tableau* $\overline{\mathbf{A}}$ *derived from* $\mathbf{A}$ *according to the general scheme (41) is HG-compatible.* ■

3.3 A digression on the case of negative weights

So far, I have sticked to the restriction (11) that HG weights are nonnegative. Now I want to discuss what happens to claim 2 when this non-negativity restriction (11) is dropped. This is useful in case we want to try to adapt to OT algorithms for HG that do not return weights that are necessarily nonnegative; a specific such case will come up in Section 4. Thus, consider the variant of the Weighting problem (33) without the non-negativity restriction, as in (42). I will denote by $\text{WP}_{\text{unr}}(\overline{\mathbf{A}})$ the instance of problem (42) corresponding to an HG-comparative tableau $\overline{\mathbf{A}}$, or equivalently the set of its solutions.

(42) *given:* an HG-compatible HG-comparative tableau $\overline{\mathbf{A}}$;
find: a weight vector $\boldsymbol{\theta}$ (with no restriction on the sign of the weights) HG-compatible with the tableau $\overline{\mathbf{A}}$.

Claim 2 establishes an equivalence between the Ranking problem (32) and the Weighting problem (33). Unfortunately, this equivalence does not extend to the unrestricted variant (42) of the latter. Here is a trivial counterexample. Consider the OT-comparative row $\mathbf{a}$ in (43). The corresponding HG-comparative row derived according to (38) is $\overline{\mathbf{a}}$ in (43). The weight vector $\boldsymbol{\theta}$ in (43) is HG-compatible with the derived HG-comparative row $\overline{\mathbf{a}}$, and has negative components. Yet, this weight vector admits the derived ranking $\gg$ in (43), which is not OT-compatible with the OT-comparative row $\mathbf{a}$.

(43)
$$\begin{array}{ccc} \mathbf{a} = [\,\text{W},\ \text{L},\ \text{L}\,] & \xleftrightarrow{\textbf{not } \text{OT}-\text{compatible}} & C_2 \gg C_3 \gg C_1 \\ \Big\downarrow \text{row derived by (38)} & & \Big\uparrow \text{ranking derived by (39)} \\ \overline{\mathbf{a}} = [\,1,\ -1,\ -1\,] & \xleftrightarrow[\text{HG}-\text{compatible}]{} & \boldsymbol{\theta} = \begin{bmatrix} \theta_1 = -4 \\ \theta_2 = -3 \\ \theta_3 = -3 \end{bmatrix} \end{array}$$

Yet, it turns out that the equivalence established by claim 2 does indeed extend to the unrestricted variant (42) of the Weighting problem, provided the OT-comparative tableau we start from is slightly pre-processed. Here are the details. Consider two comparative tableaux $\mathbf{A}$ and $\mathbf{A}'$ with the same

number n of columns but a possibly different number of rows. We say that $\mathbf{A}$ and $\mathbf{A}'$ are *OT-equivalent* iff the following condition holds: a ranking $\gg$ is OT-compatible with $\mathbf{A}$ iff it is OT-compatible with $\mathbf{A}'$. Thus, equivalent tableaux yield the same instance of the Ranking problem (32), namely $\mathrm{RP}(\mathbf{A}) = \mathrm{RP}(\mathbf{A}')$. Here is an obvious example of OT equivalence: if a row of a comparative tableau $\mathbf{A}$ contains m entries equal to L, then $\mathbf{A}$ is equivalent to the tableau $\mathbf{A}'$ obtained from $\mathbf{A}$ by replacing that row with m rows identical to it but for the fact that each of them retains only one of the m L's of the original row, while the others are replaced by E's. An example is provided in (44): the tableau $\mathbf{A}$ is equivalent to the tableau $\mathbf{A}'$ obtained by splitting a row of $\mathbf{A}$ with two L's into two rows with a single L each.

(44) $\mathbf{A} = \begin{bmatrix} & & \vdots & \\ \mathrm{E} & \mathrm{W} & \textcircled{L} & \textcircled{L} \\ & & \vdots & \end{bmatrix}$ $\mathbf{A}' = \begin{bmatrix} & & \vdots & \\ \mathrm{E} & \mathrm{W} & \textcircled{L} & \mathrm{E} \\ \mathrm{E} & \mathrm{W} & \mathrm{E} & \textcircled{L} \\ & & \vdots & \end{bmatrix}$

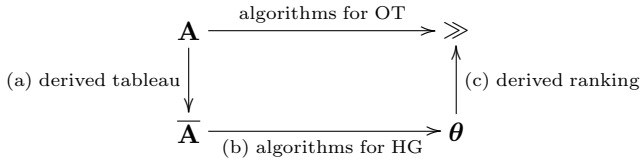
Given a comparative tableau $\mathbf{A}$, I can thus always construct an equivalent tableau $\mathbf{A}'$ such that each row of $\mathbf{A}'$ contains at most one L. In other words, I can assume without loss of generality that a given comparative tableau has at most one L per row. In this case, the equivalence between the Ranking and the Weighting problem established by claim 2 extends to the case where the non-negativity restriction (11) is dropped in the formulation of the latter. As noted in Appendix A.2, claim 5 immediately follows from the detailed proof of claim 2.

Claim 5 *Given an OT-comparative tableau $\mathbf{A}$ that has at most one L per row, consider the corresponding HG-comparative tableau $\overline{\mathbf{A}}$ derived from $\mathbf{A}$ as in (38). If $\overline{\mathbf{A}}$ is HG-compatible, then $\mathbf{A}$ is OT-compatible. More precisely, if a weight vector $\boldsymbol{\theta}$ (with possibly negative components) solves the instance $WP_{\mathrm{unr}}(\overline{\mathbf{A}})$ of the unrestricted Weighting problem (42), then any ranking derived from $\boldsymbol{\theta}$ according to (39) solves the instance $RP(\mathbf{A})$ of the Ranking problem (32).* ■

4 Algorithmic consequences

The peculiar notion of OT-compatibility (14) enforces *strict domination*, according to which the highest ranked relevant constraint "takes it all". Because of this property, OT looks *prima facie* like a rather exotic combinatorial framework. Exotic in the sense that it does not seem to have any close correspondent within core Machine Learning. For this reason, computational OT has been developed in the current literature along the lines described in (1): algorithms for OT have been developed from scratch, with no connections to methods and results from Machine Learning. This classical approach to computational OT corresponds to the top horizontal arrow in the scheme (45). As stated in (2), in order to bridge this gap between Computational OT and Machine Learning, various scholars have recently started to explore the alternative framework of HG, since HG comes with well established algorithms from the theory of linear classification. Claims 2 and 5 bear on this debate. In fact, they show that Machine Learning algorithms for HG can be "translated" into algorithms for OT according to the scheme (45): in step (45a), the OT-comparative tableau $\mathbf{A}$ given with an instance $\mathrm{RP}(\mathbf{A})$ of the Ranking problem is translated into the derived HG-comparative tableaux $\overline{\mathbf{A}}$ according to (38); in step (45b), Machine Learning algorithms for HG are used to determine a solution $\boldsymbol{\theta}$ of the corresponding Weighting problem $\mathrm{WP}(\overline{\mathbf{A}})$ or its unrestricted variant $\mathrm{WP}_{\mathrm{unr}}(\overline{\mathbf{A}})$; finally in step (45c), the weight vector $\boldsymbol{\theta}$ is translated through (39) into derived rankings that are guaranteed to solve the original Ranking problem $\mathrm{RP}(\mathbf{A})$ by claims 2 and 5. This is the new algorithmic strategy anticipated in (3). Thus, the very simple claims 2 and 5 provide computational OT with a whole new range of algorithmic techniques. In this section, I illustrate the fruitfulness of this new approach.

(45)



Computational OT has developed two types of algorithms for the Ranking problem. Given an OT-comparative tableau, *batch* ranking algorithms "work by column" and thus need to look at all comparative rows at once. *Online* ranking algorithms instead "work by row" and thus only look at a single comparative row at the time. Tesar (1995) and Tesar and Smolensky (1998) develop an efficient batch ranking algorithm, called *Recursive Constraint Demotion* (henceforth: RCD). As my first illustration of the general algorithmic strategy (45), I show in Magri (2010a) that Tesar and Smolensky's RCD for the Ranking problem in OT "corresponds" to the classical *Fourier-Motzkin Elimination Algorithm* (henceforth: FMEA) for the Weighting problem in HG; see for instance Bertsimas and Tsitsiklis (1997, pp. 70-74). In other words, the scheme in (46) holds: if we map OT-comparative tableaux into derived HG-comparative tableaux as in (38) and weight vectors into derived rankings as in (39), then RCD and the FMEA are the same algorithm.

(46)
$$\begin{array}{ccc} \mathbf{A} & \xrightarrow{\text{RCD}} & \gg \\ \text{derived tableau}\downarrow & & \uparrow\text{derived ranking} \\ \overline{\mathbf{A}} & \xrightarrow[\text{FMEA}]{} & \boldsymbol{\theta} \end{array}$$

In this section, I focus on online ranking algorithms. These are algorithms that tackle the Ranking problem $\text{RP}(\mathbf{A})$ by looking at one row of the OT-comparative tableau $\mathbf{A}$ at the time, possibly multiple times, as follows. The algorithm maintains a *current ranking*, which represents its current hypothesis of the target ranking. At each time, the algorithm is fed with a *current comparative row*, sampled from $\mathbf{A}$. If the current ranking fails to account for the current comparative row, then the algorithm takes action by updating its current ranking. As noted in Section 1, the currently most widely used OT online algorithm is Boersma; Boersma's (1997; 1998) *Gradual Learning Algorithm* (henceforth: GLA). Various studies have shown the good modeling capabilities of this simple algorithm; see for instance Boersma and Levelt (2000), Curtin and Zuraw (2002), Boersma and Hayes (2001), etcetera. Yet, the algorithm has resisted theoretical analysis and in particular its convergence has remained an open issue for many years; see Keller and Asudeh (2002) for discussion. Until the issue has been recently settled by Pater (2008), who has shown that the GLA does not converge via a simple counterexample. Thus, one of the main open questions in the current OT computational literature is the following: how should the GLA be modified so as to guarantee convergence? In this Section, I tackle this question through the algorithmic strategy (47): derived tableaux and derived rankings are used to "translate" provably convergent HG online algorithms into provably convergent OT online algorithms.

(47)
$$\begin{array}{ccc} \mathbf{A} & \xrightarrow{\text{convergent GLAs}} & \gg \\ \text{derived tableau}\downarrow & & \uparrow\text{derived ranking} \\ \overline{\mathbf{A}} & \xrightarrow[\text{HG online algorithms}]{} & \boldsymbol{\theta} \end{array}$$

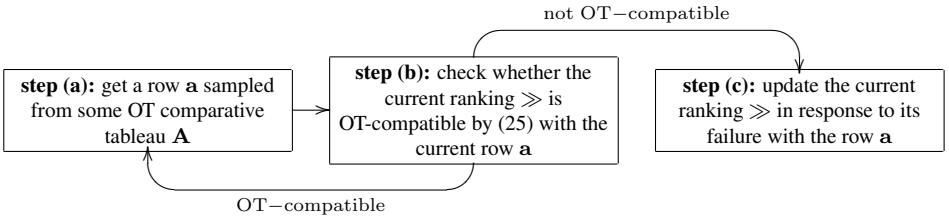
In particular, I apply this strategy (47) to the case of the *Perceptron* algorithm, a classical HG online algorithm. And I thus obtain convergence for a variant of Boersma's (non-stochastic promotion/demotion) GLA. Furthermore, I show how to derive bounds on the worst-case number of updates. Finally, I point out that the reasoning is completely general, and thus extends from the Perceptron algorithm to any online algorithm for HG (namely, any online algorithm for linear classification). These computational developments thus greatly enrich our algorithmic tools for modeling the acquisition of phonology within the mainstream phonological framework of OT.

4.1 OT and HG online algorithms

An *OT online algorithm* maintains a *current ranking*, which represents its current hypothesis on the target grammar. It initializes its current ranking to some predefined *initial ranking*. And it keeps updating its current ranking through the three steps in (48). At step (48a), the algorithm receives an OT-comparative row $\mathbf{a}$; at step (48b), the algorithm checks whether its current ranking $\gg$ is

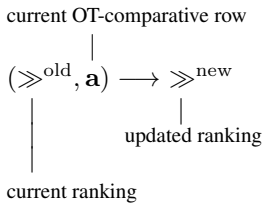
OT-compatible with this *current row* $\mathbf{a}$; if it isn't, then the algorithm takes action at step (48c), by updating its current ranking to a "slightly" modified ranking. I assume that the comparative rows fed to the algorithm at step (48a) are sampled from a fixed, given OT-compatible comparative tableau $\mathbf{A}$, called the *input tableau*. The algorithm *converges* iff it can only perform a finite number of updates for any input OT-compatible tableau. If the algorithm converges, then its final ranking solves the instance $\mathrm{RP}(\mathbf{A})$ of the Ranking problem (32) corresponding to the input tableau $\mathbf{A}$.

(48)



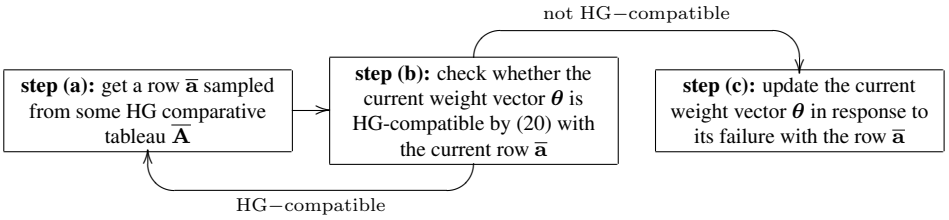
OT online algorithms differ in how they update their current ranking at step (48c). An *OT update rule* is a mapping of the form (49): it takes an OT-comparative row $\mathbf{a}$ and a current ranking $\gg^{\mathrm{old}}$ and it returns an updated ranking $\gg^{\mathrm{new}}$. An OT update rule (49) can perform one or both of two operations: it can perform constraint *demotion*, according to which some constraints are demoted to a lower rank; and/or it can perform constraint *promotion*, according to which some constraints are promoted to a higher rank.

(49)

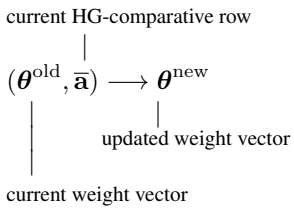


An *HG online algorithm* has an analogous shape. It maintains a *current weight vector* which represents its current hypothesis on the target grammar. It initializes its current weight vector to some predefined *initial weight vector*. And it keeps updating its current weight vector through the three steps in (50). At step (50a), the algorithm receives an HG-comparative row $\overline{\mathbf{a}}$; at step (50b), the algorithm checks whether its current weight vector $\boldsymbol{\theta}$ is HG-compatible with this *current row* $\overline{\mathbf{a}}$; if it isn't, then the algorithm takes action at step (50c), by updating its current weight vector to a "slightly" modified vector. I assume that the comparative rows fed to the algorithm at step (50a) are sampled from a fixed, given HG-compatible comparative tableau $\overline{\mathbf{A}}$, called the *input tableau*. The algorithm *converges* iff it can only perform a finite number of updates for any input HG-compatible tableau. If the algorithm converges, then its final weight vector solves the instance $\mathrm{WP}(\overline{\mathbf{A}})$ of the Weighting problem (33) corresponding to the input tableau $\overline{\mathbf{A}}$.

(50)



HG online algorithms differ in how they update their current weight vector at step (50c). An *HG update rule* is a mapping of the form (51): it takes an HG-comparative row $\overline{\mathbf{a}}$ and a current weight vector $\boldsymbol{\theta}^{\mathrm{old}}$ and it returns an updated weight vector $\boldsymbol{\theta}^{\mathrm{new}}$.

(51)


This section shows that, if OT-comparative rows are paired up with derived HG-comparative rows as in (38) and weight vectors with derived rankings as in (39), then HG update rules (51) can be translated into OT update rules (49) in such a way that convergence of the HG online algorithm (50) translates into convergence of the OT online algorithm (48). From this perspective, it makes sense to slightly relax the definition of HG. In section 2, I have introduced the restriction (11) that the weights be nonnegative. This means that HG update rules (51) need to preserve the non-negativity of the weights. This is not trivial.[5] Throughout this subsection, I will thus slightly relax the definition of HG presented in section 2, by dropping the restriction (11) that the weights be nonnegative.

(52) ~~$\theta_1 \geq 0, \ldots, \theta_n \geq 0$~~

HG update rules are very well studied in the field of linear classification; see for instance Cesa-Bianchi and Lugosi (2006, Chp. 12) for a modern introduction. As an example, consider the classical HG update rule (53): the updated weight vector $\boldsymbol{\theta}^{\text{new}} = (\theta_1^{\text{new}}, \ldots, \theta_n^{\text{new}})$ is obtained by adding component by component the current HG-comparative row $\overline{\mathbf{a}} = (\overline{a}_1, \ldots, \overline{a}_n)$ to the current weight vector $\boldsymbol{\theta}^{\text{old}} = (\theta_1^{\text{old}}, \ldots, \theta_n^{\text{old}})$. Note that this HG update rule only makes sense because I have relaxed the nonnegativity requirement on the weights by (52), as there is no guarantee that the weights will remain positive, even if we start from large positive initial weights. The HG online algorithm (50) with the update rule (53) is known as the *Perceptron* algorithm in the Machine Learning literature. The classical convergence claim 6 holds; its proof is recalled in Appendix A.3.

(53) $\theta_k^{\text{new}} = \theta_k^{\text{old}} + \overline{a}_k$

Claim 6 *The HG online algorithm (50) with the Perceptron HG update rule (53) converges, provided that the input comparative tableau is HG-compatible.* ■

In this section, I will show how the convergent Perceptron HG update rule (53) can be turned into a convergent OT update rule that performs both promotion and demotion.

4.2 Restatement of OT online algorithms in terms of weight vectors

As noted in section 3, we can represent rankings through numerical weight vectors: we say that a ranking $\gg$ is *derived* from a given weight vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ iff it satisfies condition (39) for any pair of constraint C_h, C_k, repeated in (54). This condition says that the ranking $\gg$ respects the ordering of the constraints that is implicit in the relative size of their weights $\theta_1, \ldots, \theta_n$.

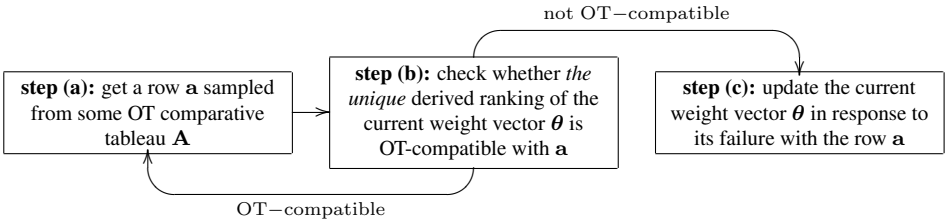
(54) $\theta_h > \theta_k \Longrightarrow C_h \gg C_k$

Let me thus introduce a slight variant of the definition (48) of the OT online algorithm, whereby the algorithm maintains at every time a representation of the current ranking in terms of a weight vector, rather than the current ranking itself. To get started, suppose that the current weight vector entertained by the algorithm at every iteration happens to have pairwise distinct components. Then it admits a unique derived ranking. Namely, it represents a unique current ranking. If we could restrict ourselves to current weight vectors that have pair-wise distinct components, then we could restate the OT online algorithm (48) as in (55).[6]

[5] The recent HG computational literature usually tries to get around this problem by starting out with large positive weights; see for instance Jesney and Tessier (2009). But this trick does not guarantee that the weights will stay non-negative at every iteration until convergence, as the number of updates depends on the size of the initial weights. An alternative more principled approach is to use HG online algorithms with a multiplicative rather than additive update rule; see Magri (2007).

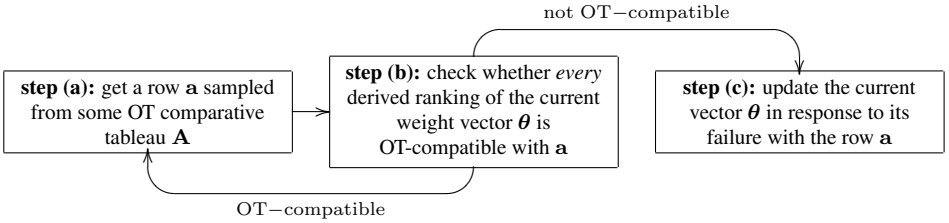
[6] Indeed, it is not hard to enforce the current weights to be always pair-wise distinct. For instance, we could select an initial weight vector with pair-wise distinct *fractional* components, and then allow the update rule to only add or subtract to the current weights an *integer* amount.

(55)



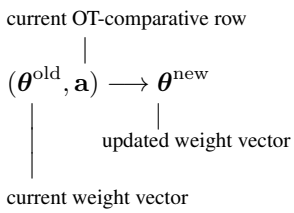
Consider now the case where the current weight vector might happen to have two or more identical components, thus admitting multiple derived rankings. How should we proceed in this case? There are two possible ways to go. One way to go is to check at step (55b) whether *some* ranking derived from the current weight vector is OT-compatible with the current OT-comparative row, and to update only in case none of the derived rankings is OT-compatible. Another way to go is to check at step (55b) whether *all* rankings derived from the current weight vector are OT-compatible with the current OT-comparative row, and to update provided that even just one of them isn't OT-compatible. Suppose we go the first way. Then, if the algorithm converges after a finite number of updates, it will return a weight vector with the property that *some* of its derived rankings are OT-compatible with the input OT-comparative tableau. But that is not very useful: how do we decide for a given derived ranking whether it is what we want or not? Thus, we go the other way, namely we require at step (55b) that *every* ranking derived from the current weight vector be OT-compatible with the current OT-comparative row, as in the restatement in (56). From now on, I will refer to the scheme in (56) as the *OT online algorithm*.

(56)



The idea of representing the current ranking entertained by an OT online algorithm as a weight vector through the notion of derived ranking is due to Boersma (1997, 1998, 2008). I think it is an extremely important idea. In particular, it is important because it allows OT update rules to manipulate weight vectors rather than rankings, as initially assumed in (49). More explicitly, we can now re-define an OT update rule to be used in step (56c) as a mapping of the form (57): it takes an OT-comparative row $\mathbf{a}$ together with the current weight vector $\boldsymbol{\theta}^{\mathrm{old}}$ and it returns an updated weight vector $\boldsymbol{\theta}^{\mathrm{new}}$.

(57)



Once OT online algorithms are restated in terms of weight vectors, OT and HG online algorithms only differ under two obvious respects. First, OT online algorithms are fed with OT-comparative rows while HG online algorithms are fed with HG-comparative rows. Second, OT online algorithms check for OT-compatibility, while HG online algorithms check for HG-compatibility. Let me pause and illustrate the revised notion of OT online algorithm (56) in terms of weight vectors with a few examples taken from the literature.

4.3 A quick review of the literature on OT online algorithms

Let me say that a constraint C_k is *currently* loser-preferrer (winner-preferrer) iff the corresponding entry a_k in the OT-comparative row $\mathbf{a} = [a_1, \ldots, a_n]$ currently fed to the OT online algorithm is an

L (a W, respectively). Consider the current OT-comparative row (58a) and the current weight vector (58b). The two constraints C_4 and C_6 are both currently loser-preferrers, as they both have an L in the current comparative row. Yet, there is a crucial difference between them. The current weight $\theta_6^{\text{old}} = 5$ of constraint C_6 is smaller than the current weight $\theta_1^{\text{old}} = 10$ of the winner-preferrer C_1. The current weight $\theta_4^{\text{old}} = 15$ of constraint C_4 is instead larger than the current weights $\theta_1^{\text{old}} = 10$ and $\theta_2^{\text{old}} = 5$ of both winner-preferrers C_1 and C_2. This difference between the two loser-preferrers C_4 and C_6 is important enough to warrant a name: a loser-preferrer C_ℓ is *currently undominated* iff there is no current winner-preferrer C_k currently ranked above C_ℓ (in the sense that $\theta_k^{\text{old}} > \theta_\ell^{\text{old}}$). Thus, C_4 is currently undominated, while C_6 is not.

(58) a. $\mathbf{a} =$

C_1	C_2	C_3	C_4	C_5	C_6
W	W	E	(L)	E	(L)

b. $\boldsymbol{\theta}^{\text{old}} =$

C_1	C_2	C_3	C_4	C_5	C_6
10	5	20	(15)	100	(5)

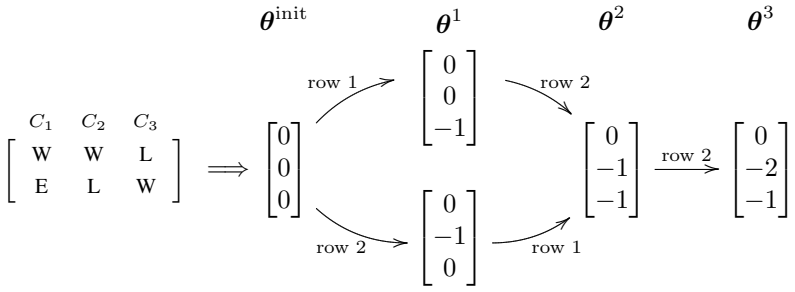
An important example of OT update rule is (59), that demotes by 1 all currently undominated loser-preferrers. Boersma (1998, p. 323-327) notes that Tesar and Smolensky's (1998) analysis straightforwardly applies to the convergence of the OT online algorithm (56) with this update rule (59).[7]

(59) Demote by 1 each currently undominated loser-preferring constraint.

$$\theta_k^{\text{new}} = \begin{cases} \theta_k^{\text{old}} - 1 & \text{if } C_k \text{ is a currently undominated loser-preferrer} \\ \theta_k^{\text{old}} & \text{otherwise} \end{cases}$$

The behavior of the OT online algorithm with the update rule (59) is illustrated in (60). The algorithm starts from the null initial vector. At the first iteration, the algorithm can receive either the first or the second row of the input comparative tableau. Suppose that it receives the first row. Since the current weight vector $\boldsymbol{\theta}^{\text{init}}$ admits derived rankings (such as $C_3 \gg C_2 \gg C_1$) not OT-compatible with that row, then the algorithm updates $\boldsymbol{\theta}^{\text{init}}$ to $\boldsymbol{\theta}^1$ by demoting by 1 the weight of the current loser-preferrer C_3. At the next iteration, the algorithm can again receive either the first or the second row of the input comparative tableau. Since all derived rankings of the current weight vector $\boldsymbol{\theta}^1$ are OT-compatible with the first row, nothing happens if the algorithm receives the first row. Since instead the current weight vector $\boldsymbol{\theta}^1$ admits derived rankings not OT-compatible with the second row (such as $C_2 \gg C_1 \gg C_3$), once the algorithm receives the second row, it updates its current weight vector $\boldsymbol{\theta}^1$ to $\boldsymbol{\theta}^2$ by demoting by 1 the weight of the current loser-preferrer C_2. And so on. No matter the order that the comparative rows are fed to the algorithm, after three updates the algorithm entertains the weight vector $\boldsymbol{\theta}^3$, whose unique derived ranking $C_1 \gg C_3 \gg C_2$ is OT-compatible with the input tableau.

(60)



The update rule (59) only performs demotion, since it only shrinks the weights of currently undominated loser-preferrers. Yet, various authors have noted that demotion-only is not sufficient and that we do want update rules for the OT online algorithm that perform promotion too. For non-computational arguments in favor of constraint promotion, see Gnanadesikan (2004), Stemberger and Bernhardt (1999), Bernhardt and Stemberger (1998), and Stemberger et al. (1999) among others. An explicit computational argument for constraint promotion is due to Boersma (1997): constraint demotion alone is not able to model certain cases of learning in the presence of variation. Another computational argument for constraint promotion in a non-stochastic setting is due to Magri (2010b): constraint demotion alone is not able to model Hayes's (2004) "early stage" of the

[7] Thanks to B. Tesar (p.c) for clarification on this point.

acquisition of phonotactics, when the learner has no access to alternations and thus can only posit fully faithful underlying forms. These considerations motivate the search for provably convergent update rules (57) for the OT online algorithm (56) that perform promotion too. This is not an easy task. Tesar and Smolensky (1998) explicitly warn against promotion, because of Dresher's (1999) *credit problem*: how can we determine which current winner-preferrers should be promoted, so that we avoid promoting constraints that in the end might have to sit at the bottom of the ranking? Boersma (1997) conjectures that we might be able to get around the credit problem by performing small, gradual updates at each iteration: if we promote by a small amount, then we might expect that little harm will come from promoting the wrong constraints. He thus suggests the update rule (61), that promotes (demotes) current winner-preferrers (undominated loser-preferrers) by a small amount, say 1. The OT online algorithm (56) with this update rule (61) is called the (deterministic) *Gradual Learning Algorithm* (henceforth: GLA).

(61) Promote (demote) each current winner-preferrer (undominated loser-preferrer) by 1.

$$\theta_k^{\text{new}} = \begin{cases} \theta_k^{\text{old}} + 1 & \text{if } C_k \text{ is currently winner-preferrer} \\ \theta_k^{\text{old}} - 1 & \text{if } C_k \text{ is a currently undominated loser-preferrer} \\ \theta_k^{\text{old}} & \text{otherwise} \end{cases}$$

As stressed in Keller and Asudeh (2002), convergence of Boersma's GLA has remained an open issue for a few years. Until Pater (2008) has shown that convergence does not hold in the general case: for instance, the algorithm does not converge in the case of the input OT-comparative tableau (62), where the credit problem is exacerbated by stacking rows with two winner-preferrers. See Magri (2010b) for a detailed explanation of Pater's counterexample.

(62)

C_1	C_2	C_3	C_4	C_5
W	L	W		
	W	L	W	
		W	L	W
			W	L

The problem of devising provably convergent promotion/demotion update rules for the OT online algorithm (56) is thus currently open. How can Boersma's update rule (61) be modified, so that we can guarantee convergence despite promotion?

4.4 A convergent promotion/demotion update rule: case of a unique loser-preferrer

To simplify the discussion, let me temporarily make assumption (63). As noted in subsection 3.3, assumption (63) is not really restrictive, as the input tableau can always be pre-processed in such a way to have a unique L per row. In any case, assumption (63) is only temporary, and will be dropped in subsection 4.6.

(63) The rows fed to the OT online algorithm at step (56a) only have a unique L.

Let me distinguish two cases, depending on whether the current OT-comparative row contains a unique W, as in (64a); or else contains multiple W's, say two as in (64b). These two cases are intuitively very different. The former case (64a) with a unique winner-preferrer is simple, because it raises no credit problem: we know that the unique winner-preferer must in the end be ranked above the loser-preferrer, irrespectively of the rest of the input comparative tableau. The latter case (64b) with two winner-preferrers is more delicate, because it raises a credit problem: we don't know which one of the two winner-preferrers needs in the end to be ranked *above* the loser-preferrer, as one of them might actually need to be ranked in the end *below* the loser-preferrer, depending on what the rest of the input comparative tableau looks like.

(64) a.

...	...	C_k	...	C_ℓ	...
...	...	W	...	L	...

b.

...	C_h	...	C_k	...	C_ℓ	...
...	W	...	W	...	L	...

Boersma's promotion-demotion update rule (61) treats the two cases in (64) in the same way: any winner-preferrer gets promoted by 1, no matter whether it appears in a "simple" row (64a) with a unique winner-preferer or in a "challenging" row (64b) with multiple winner-preferrers. This does not look like a good idea though, since it does not capture the intrinsic logic of OT, namely the crucial difference just discussed between the two cases in (64). I thus suggest the following more principled alternative. In the case of the "simple" comparative row (64a) with a unique winner-preferrer, we can confidently promote that unique winner-preferer by the same amount we demote the loser-preferrer, say 1. But in the case of the "challenging" comparative row (64b) with two winner-preferrers, we should be *cautious* and split our confidence between the two winner-preferrers, by promoting each one just by $1/2$. In the general case, the uncertainty "scales" with the total number w of winner-preferrers, and we should thus promote each winner-preferer just by $1/w$. In conclusion, I suggest the new *cautious* promotion/demotion OT update rule (65).[8]

(65) Promote each current winner-preferrer by $1/w$; demote the unique loser-preferrer by -1

$$\theta_k^{\text{new}} = \begin{cases} \theta_k^{\text{old}} + \frac{1}{w} & \text{if } C_k \text{ is currently winner-preferrer} \\ \theta_k^{\text{old}} - 1 & \text{if } C_k \text{ is the unique current loser-preferrer} \\ \theta_k^{\text{old}} & \text{otherwise} \end{cases}$$

where w is the total number of current winner-preferrers.

Of course, I get an equivalent update rule by multiplying both the promotion amount $1/w$ and the demotion amount -1 in (65) by the same positive constant.[9] In particular, by multiplying by the total number w of winner-preferrers, I get the equivalent update rule (66), with integer promotion and demotion amounts.

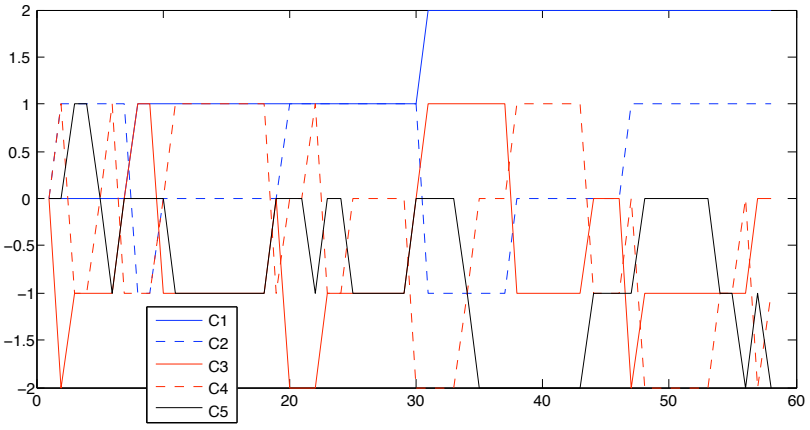
(66) Promote each current winner-preferrer by 1; demote the unique loser-preferrer by $-w$

$$\theta_k^{\text{new}} = \begin{cases} \theta_k^{\text{old}} + 1 & \text{if } C_k \text{ is currently winner-preferrer} \\ \theta_k^{\text{old}} - w & \text{if } C_k \text{ is the unique loser-preferrer} \\ \theta_k^{\text{old}} & \text{otherwise} \end{cases}$$

where w is the total number of current winner-preferrers.

To illustrate, I give in (67) the dynamics over time of the weights entertained by the OT online algorithm with the new cautious OT update rule (66) run on Pater's input tableau (62) with the rows sampled uniformly. The algorithm converges, but the dynamics of the weights looks very complicated, as they oscillate up and down before they settle on their final value. Given this complicated non-monotonic dynamics that we get with promption-demotion update rules, how can we analytically prove convergence? Here is a proof of convergence, based on the results of Section 3.

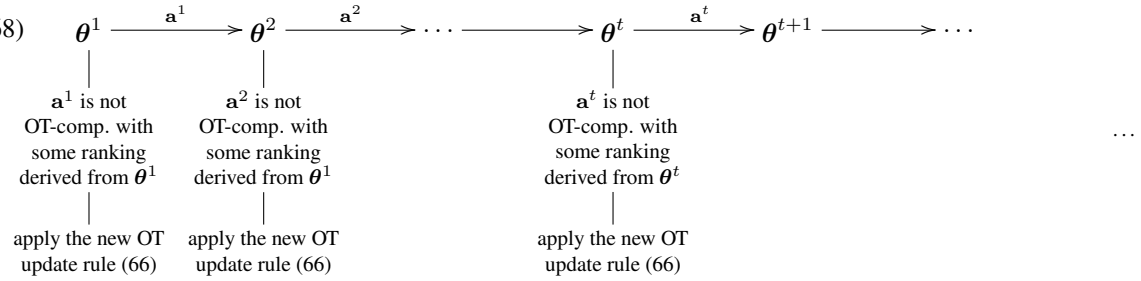
(67)



[8] Since the current comparative row contains a unique L by assumption (63), than the corresponding loser-preferrer has got to be currently undominated in order for that row not to be OT-compatible with some ranking derived from the current weight vector, and thus trigger an update.

[9] *Equivalent* in the sense that the original update rule and the rescaled update rule yield exactly the same learning path, provided that the initial weights are all identical. Thanks to Paul Smolensky for discussion on this point.

Suppose by contradiction that the OT online algorithm (56) with the new cautious promotion/demotion update rule (66) does not converge. This means that there exists an input OT-compatible comparative tableau $\mathbf{A}$ such that we can construct the infinite sequence in (68): at each time t, we can sample from the input tableau $\mathbf{A}$ an OT-comparative row (denote it by $\mathbf{a}^t$) that happens not to be OT-compatible with some ranking derived from the current weight vector (denote it by $\boldsymbol{\theta}^t$) and thus forces the OT online algorithm to update to a slightly different weight vector (denote it by $\boldsymbol{\theta}^{t+1}$) using the update rule (66).

(68)


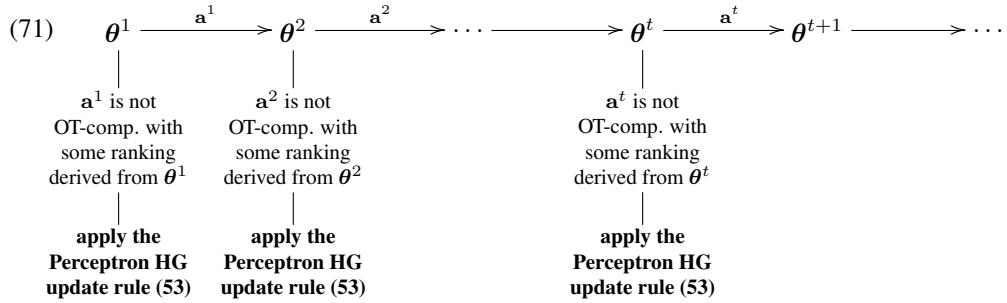
The new cautious promotion/demotion OT update rule (66) looks intuitively related to the mapping (38) used in Section 3 to turn OT- into derived HG-comparative rows, repeated in (69): a W is replaced with $+1$ and an L with $-w$, where w is the total number of W's in the given OT-comparative row. Let $\overline{\mathbf{a}}^t$ be the HG-comparative row derived according to (69) from the current OT-comparative row $\mathbf{a}^t$ fed to the algorithm at time t in the run (68).

(69)
$$\mathbf{a} = [a_1, \ldots, a_n]$$
$$\downarrow$$
$$\overline{\mathbf{a}} = [\overline{a}_1, \ldots, \overline{a}_n] \qquad \text{such that } \overline{a}_k \doteq \begin{cases} -w & \text{if } a_k = \text{L} \\ 0 & \text{if } a_k = \text{E} \\ 1 & \text{if } a_k = \text{W} \end{cases}$$

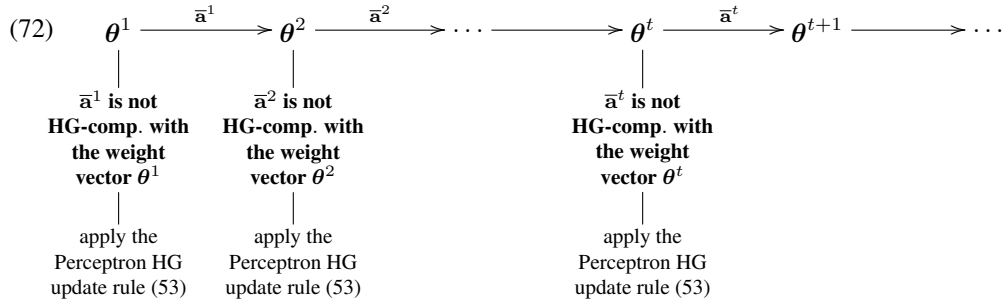
To bring out the connection between the new cautious update rule (66) and the mapping (69), consider the example in (70). Suppose that the current OT-comparative row is $\mathbf{a}^t = [\text{W}, \text{W}, \text{E}, \text{L}]$, with two W's and a unique L. Suppose that the current weight vector is $\boldsymbol{\theta}^t = [1, 1, 1, 1]$. Let's update the current weight vector in response to the current comparative row according to the new cautious promotion/demotion OT update rule (66). The current weights of the two winner-preferrers are increased by 1 while the weight of the loser-preferrer is decreased by 2. The updated weight vector is therefore $\boldsymbol{\theta}^{t+1} = [2, 2, 1, -1]$. This update can equivalently be described as follows. The current comparative row $\mathbf{a}^t = [\text{W}, \text{W}, \text{E}, \text{L}]$ can be mapped through (69) to the derived HG-comparative row $\overline{\mathbf{a}}^t = [1, 1, 0, -2]$. Let's update the current weight vector $\boldsymbol{\theta}^t = [1, 1, 1, 1]$ in response to this HG-comparative row according to the Perceptron HG-update rule (53). The HG-comparative row thus gets added to the current weight vector. The updated weight vector thus obtained is again $\boldsymbol{\theta}^{t+1} = [2, 2, 1, -1]$.

(70)
$$\Big(\mathbf{a}^t = [\text{W}, \text{W}, \text{E}, \text{L}],\ \boldsymbol{\theta}^t = [1, 1, 1, 1]\Big) \xrightarrow{\text{new OT}-\text{update rule (66)}} \boldsymbol{\theta}^{t+1} = [2, 2, 1, -1]$$
$$(69)\downarrow$$
$$\Big(\overline{\mathbf{a}}^t = [1, 1, 0, -2],\ \boldsymbol{\theta}^t = [1, 1, 1, 1]\Big) \xrightarrow[\text{Perceptron HG}-\text{update rule (53)}]{} \boldsymbol{\theta}^{t+1} = [2, 2, 1, -1]$$

The example (70) makes a general point: update according to the new cautious promotion/demotion OT update rule (66) in response to the OT-comparative row $\mathbf{a}^t$ is equivalent to update according to the Perceptron HG-update rule (53) in response to the corresponding HG-comparative row $\overline{\mathbf{a}}^t$ derived through (69). The situation (68) thus entails (71).

(71)


Given an OT-comparative tableau with at most one L per row, consider the corresponding HG-comparative tableau derived according to (69). Recall that claim 5 from Section 3 ensures that, if the derived tableau is HG-compatible with some weight vector, then the original OT-comparative tableau is OT-compatible with every ranking derived from that weight vector. This claim now plays a crucial role. In fact, the current OT-comparative row $\mathbf{a}^t$ fed to the algorithm in (71) has a unique L, by assumption (63). Furthermore, $\mathbf{a}^t$ is not OT-compatible with some ranking derived from the current weight vector $\boldsymbol{\theta}^t$. Thus, claim 5 ensures by contraposition that the corresponding derived HG-comparative row $\overline{\mathbf{a}}^t$ is not HG-compatible with the current weight vector $\boldsymbol{\theta}^t$. The situation (71) thus entails (72).

(72)


Let $\overline{\mathbf{A}}$ be the HG-comparative tableau derived from the input OT-comparative tableau according to (69). Claim 4 from Section 3 ensures that $\overline{\mathbf{A}}$ is HG-compatible, since it is derived from an OT-compatible tableau. The diagram in (72) thus contradicts claim 6, that ensures convergence for the Perceptron HG update rule (53). We have thus proved the following claim 7. It says that the new cautious promotion/demotion OT update rule (66) is immune to counterexamples such as Pater's tableau (62) against Boersma's original update rule (61), that force the algorithm to make an infinite number of updates.

Claim 7 *The OT online algorithm (56) with the new cautious promotion/demotion OT update rule (66) converges for every input OT-compatible tableau that has a unique* L *per row.* ■

4.5 Worst case number of updates

In the preceding subsection, I have concentrated on the issue of convergence. An important related issue is that of the worst case number of updates required for convergence. For instance, in the preceding subsection I have shown that the OT online algorithm with the new cautious promotion/demotion OT update rule (66) converges in the case of Pater's comparative tableau (62), contrary to the case of Boersma's update rule (61). But what is the worst-case number of updates that the algorithm will make to reach convergence? The algorithmic approach developed in the preceding subsection yields straightforward bounds on the worst-case number of updates. To keep things simple, let me stick to assumption (63) that input tableau $\mathbf{A}$ has a unique L per row. Consider a run (73a) of the OT online algorithm. At each time t, the algorithm is fed with a row $\mathbf{a}^t$ sampled from $\mathbf{A}$; without loss of generality, assume that $\mathbf{a}^t$ is not OT-compatible with some ranking derived from the current weight vector $\boldsymbol{\theta}^t$; thus, the latter is updated to the weight vector $\boldsymbol{\theta}^{t+1}$ according to the new cautious promotion/demotion OT update rule (66). This procedure is repeated until the

OT online algorithm entertains a weight vector whose derived rankings are all OT-compatible with the input tableau $\mathbf{A}$, so that no further update can be triggered. Let me denote the time at which this happens by $T_{\text{OT}} = T_{\text{OT}}(\mathbf{A})$. As noted in the preceding Subsection, the same learning path can be re-described in terms of the HG online algorithm with the Perceptron HG update rule, as in (73b). At each time t, the algorithm is fed with the HG-comparative row $\overline{\mathbf{a}}^t$ derived through (69) from the OT-comparative row $\mathbf{a}^t$ fed to the OT online algorithm at time t; by claim 5, the current HG-comparative row $\overline{\mathbf{a}}^t$ is not HG-compatible with the current weight vector $\boldsymbol{\theta}^t$; and it thus triggers an update of the current weight vector according to the Perceptron HG update rule (53). Note that the weight vector entertained at time T_{OT} in the general case need not be HG-compatible with the derived input tableau $\overline{\mathbf{A}}$ and that further HG updates might be needed before the HG online algorithm entertains a weight vector HG-compatible with the derived input tableau $\overline{\mathbf{A}}$, so that no further HG update can be triggered. Let me denote the time at which that happens by $T_{\text{HG}} = T_{\text{HG}}(\overline{\mathbf{A}})$.

(73) a. $\boldsymbol{\theta}^{\text{init}} \longrightarrow \cdots \longrightarrow \boldsymbol{\theta}^t \xrightarrow{\mathbf{a}^t} \boldsymbol{\theta}^{t+1} \longrightarrow \cdots \longrightarrow \boldsymbol{\theta}^{T_{\text{OT}}}$

not OT-comp.

apply new OT update rule (66)

b. $\boldsymbol{\theta}^{\text{init}} \longrightarrow \cdots \longrightarrow \boldsymbol{\theta}^t \xrightarrow{\overline{\mathbf{a}}^t} \boldsymbol{\theta}^{t+1} \longrightarrow \cdots \longrightarrow \boldsymbol{\theta}^{T_{\text{OT}}} \longrightarrow \cdots \longrightarrow \boldsymbol{\theta}^{T_{\text{HG}}}$

not HG-comp.

apply HG Perceptron update rule (53)

Crucially, every time the OT online algorithm performs an OT update in (73a), the HG online algorithm performs an update too in the corresponding run (73b). Thus, the learning path described by the HG online algorithm to reach convergence is at least as long as that described by the OT online algorithm. In other words, the number T_{OT} of OT updates in (73a) is upper bounded by the number T_{HG} of HG updates in (73b). We have thus obtained claim 8. Thus, promotion/demotion OT online algorithms are not slower than classical HG online algorithms in the worst case.

Claim 8 *Consider an OT-compatible input tableau* $\mathbf{A}$ *with a unique* L *per row; let* $\overline{\mathbf{A}}$ *be the corresponding HG-comparative tableau derived through (69). The worst-case number of updates* $T_{\text{OT}}(\mathbf{A})$ *performed by the OT online algorithm run on* $\mathbf{A}$ *with the new cautious promotion/demotion OT update rule (66) is at most equal to the worst-case number of updates* $T_{\text{HG}}(\overline{\mathbf{A}})$ *performed by the HG online algorithm on the derived tableau* $\overline{\mathbf{A}}$ *with the Perceptron HG update rule (53).* ■

Let me illustrate claim 8 with a couple of concrete applications. Following Riggle (2009), let a *diagonal tableau of order* n be the comparative tableau with $n-1$ rows and n columns, whose kth row has all entries equal to E but for the kth entry which is a W and the following entry which is an L. To illustrate, I give in (74) the diagonal tableaux of order $n = 4, 5, 6$.

(74)
$$\begin{bmatrix} \text{W} & \text{L} & & \\ & \text{W} & \text{L} & \\ & & \text{W} & \text{L} \end{bmatrix}, \begin{bmatrix} \text{W} & \text{L} & & & \\ & \text{W} & \text{L} & & \\ & & \text{W} & \text{L} & \\ & & & \text{W} & \text{L} \end{bmatrix}, \begin{bmatrix} \text{W} & \text{L} & & & & \\ & \text{W} & \text{L} & & & \\ & & \text{W} & \text{L} & & \\ & & & \text{W} & \text{L} & \\ & & & & \text{W} & \text{L} \end{bmatrix}$$

Furthermore, let *Pater's tableau of order* n be the comparative tableau with $n-1$ rows and n columns obtained from the diagonal tableau by "adding" a W at the right of every L. The case with $n = 5$ was already considered in (62). To illustrate, I give in (75) Pater's tableaux of order $n = 4, 5, 6$.

(75)
$$\begin{bmatrix} \text{W} & \text{L} & \text{W} & \\ & \text{W} & \text{L} & \text{W} \\ & & \text{W} & \text{L} \end{bmatrix}, \begin{bmatrix} \text{W} & \text{L} & \text{W} & & \\ & \text{W} & \text{L} & \text{W} & \\ & & \text{W} & \text{L} & \text{W} \\ & & & \text{W} & \text{L} \end{bmatrix}, \begin{bmatrix} \text{W} & \text{L} & \text{W} & & & \\ & \text{W} & \text{L} & \text{W} & & \\ & & \text{W} & \text{L} & \text{W} & \\ & & & \text{W} & \text{L} & \text{W} \\ & & & & \text{W} & \text{L} \end{bmatrix}$$

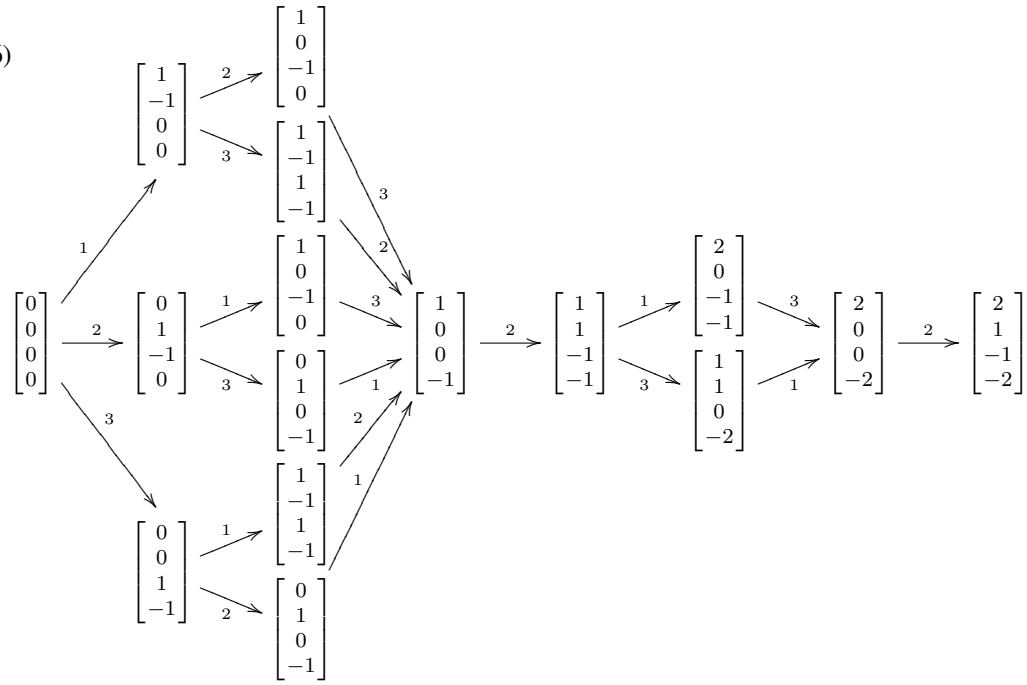
Claim 8 ensures that bounds on the worst-case number of updates T_{OT} of the OT online algorithm with the new cautious promotion/demotion OT update rule (66) can be obtained through the bounds on the worst case number of updates T_{HG} of the HG online algorithm with the HG Perceptron update rule. In Appendix A.4, I exploit this strategy in order to prove the following claims 9 and 10.

Claim 9 *The worst-case number of updates performed by the OT online algorithm (56) with the new cautious promotion-demotion OT update rule (66) run on the diagonal comparative tableau of order n starting from the null initial vector is bound by $n(n^2-1)/6$.* ■

Claim 10 *The worst-case number of updates of the OT online algorithm (56) with the new cautious promotion-demotion update rule (79) run on Pater's comparative tableau of order n starting from the null initial vector grows with n at most at the order of n^5.* ■

Bounds on the worst case number of updates obtained through this strategy are not tight. For instance, the bound provided by claim 9 for the case of the diagonal tableau of order $n = 4$ is $4(4^2-1)/6 = 10$. Yet, diagram (76) displays all possible learning paths that the OT online algorithm can walk through, showing that the algorithm only performs 7 updates (the labels "1", "2" and "3" on the arrows say which one of the three rows of the input tableau is triggering the update).

(76)



The looseness of the bounds thus obtained comes from two facts. On the one hand, the corresponding bounds on the worst-case number of HG updates might not be tight to start with. On the other hand, the HG online algorithm might in principle require further updates in order to reach HG convergence, after the OT online algorithm has already reached convergence, as sketched in (73).

4.6 A new convergent promotion/demotion update rule: the general case

In Subsection 4.4, I have assumed that the rows fed to the OT online algorithm have only one L, by (63). Let me make explicit why I had to make this restrictive assumption. The HG online algorithm (50) run with the HG Perceptron update rule (53) does not ensure that the current weights are nonnegative. Even if the algorithm is initialized with very large positive initial weights, there is no guarantee that the weights will stay positive until convergence, as the total number of updates depends on the initial weight vector. Since I cannot guarantee that the weights be nonnegative, then I cannot use claim 2 in order to prove the crucial fact that OT-incompatibility between the current OT-comparative row and the current weight vector entails HG-incompatibility between the derived

HG-comparative row and the current weight vector. Thus, I restricted myself to rows with a unique L in order to apply claim 5 instead, that does not require the weights to be non-negative. As shown by the counterexample in (43), the restriction to rows with a unique L is really crucial when the weights can be negative. In this Subsection, I show how to get around the difficulty just highlighted, and thus drop the restrictive assumption (63) that the input rows have a unique L. To get started, consider a generic current OT-comparative row (77), with possibly multiple currently undominated L's and multiple W's. How should we update in response to this OT-comparative row? I suggest the following heuristic reasoning.

(77)

	C_h	C_k		C'	C''	
[...	W	W	...	L	L	...]

For concreteness, suppose that the constraints C' and C'' are the only two currently undominated loser-preferrers in row (77). Consider then the two rows (78), that only differ from the original row (77) because of the fact that each of them keeps only one of the two L's of the original row, while the other is replaced by an E. As already noted in Subsection 3.3, the original row (77) is *OT-equivalent* to the two rows (78), in the sense that a ranking is OT-compatible with the former iff it is OT-compatible with the latter. In other words, the two L's of the original row can be split over two rows.

(78)

	C_h	C_k		C'	C''	
[...	W	W	...	L	E	...]
	C_h	C_k		C'	C''	
[...	W	W	...	E	L	...]

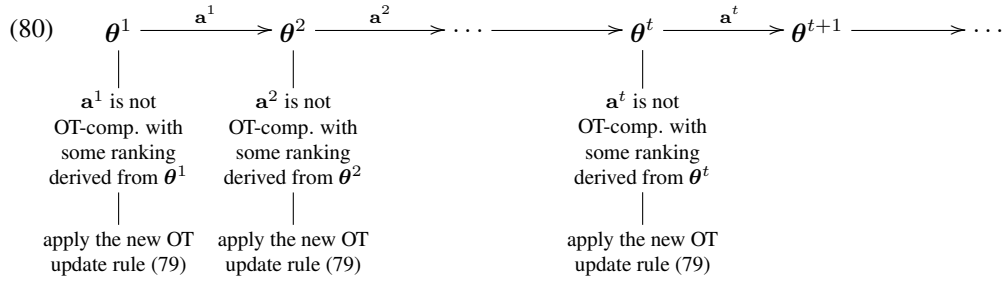
Because of this equivalence, it makes sense to define the update triggered by the original row (77) as two consecutive updates triggered by the two rows (78). Update by the first row in (78) can be performed according to the update rule (66) devised in Subsection 4.4, since that row has a unique loser-preferrer: each winner-preferrer is promoted by 1 and the unique loser-preferrer is demoted by the total number of winner-preferrers. Subsequent update by the second row in (78) can again be performed according to (66): each winner-preferrer is promoted again by 1 and the unique loser-preferrer is demoted by the total number of winner-preferrers. In conclusion, each winner-preferrer gets promoted by the total number of undominated loser-preferrers (as it gets promoted by 1 for as many times as there are undominated loser-preferrers) and each undominated loser-preferrer gets demoted by the total number of winner-preferrers. This new update rule (79) generalizes the update rule (66) considered in the preceding Section to the case of input rows with an arbitrary number of undominated loser-preferrers. In the rest of this Subsection, I show how the proof of convergence presented in Subsection 4.4 for the special case of rows with a unique L, can be extended to the general case (79).

(79) Promote each current winner-preferrer by ℓ and demote each currently undominated loser-preferrer by $-w$

$$\theta_k^{\text{new}} = \begin{cases} \theta_k^{\text{old}} + \ell & \text{if } C_k \text{ is currently winner-preferrer} \\ \theta_k^{\text{old}} - w & \text{if } C_k \text{ is a currently undominated loser-preferrer} \\ \theta_k^{\text{old}} & \text{otherwise} \end{cases}$$

where w is the total number of current winner-preferrers and ℓ is the total number of currently undominated loser-preferrers.

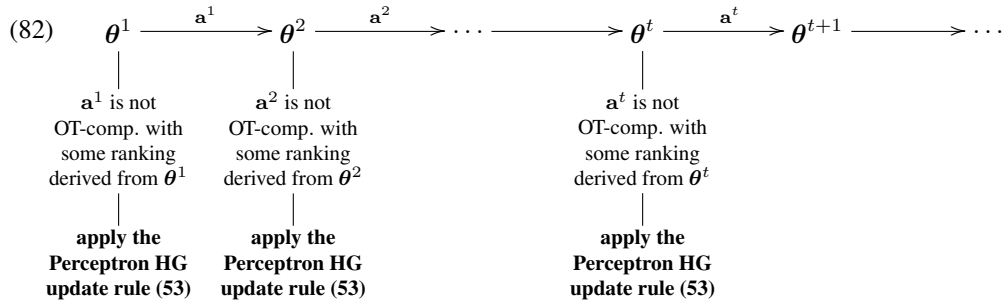
Suppose by contradiction that the OT online algorithm (56) with the new promotion/demotion update rule (79) does not converge. This means that there exists an input OT-compatible comparative tableau $\mathbf{A}$ such that we can construct the infinite sequence in (80): at each time t, we can sample from the input tableau $\mathbf{A}$ an OT-comparative row (denote it by $\mathbf{a}^t$) that happens not to be OT-compatible with some ranking derived from the current weight vector (denote it by $\boldsymbol{\theta}^t$) and thus forces the OT online algorithm to update to a slightly different weight vector (denote it by $\boldsymbol{\theta}^{t+1}$) using the new update rule (79).

(80)


Consider the mapping (81) that takes an OT-comparative row $\mathbf{a}$ and a weight vector $\boldsymbol{\theta}$ and returns a *derived* HG-comparative row $\overline{\mathbf{a}}$. Let me unpack this definition. A winner-preferrer (loser-preferrer) is a constraint C_k whose corresponding entry in the OT-comparative row $\mathbf{a} = [a_1, \ldots, a_n]$ is $a_k =$ W ($a_k =$ L, respectively). A loser-preferrer C_k is called undominated w.r.t. the weight vector $\boldsymbol{\theta}$ provided that there is no winner-preferrer with a larger weight than C_k. Let w be the number of winner-preferrers and ℓ be the number of undominated loser-preferrers. According to (81), the W of winner-preferrers are mapped to the number ℓ; and the L's corresponding to undominated loser-preferrers are mapped to the number $-w$. Note the crucial difference between the mapping (69) used in Subsection 4.4 and the mapping (81) used here: in the former case, HG-comparative rows are derived just from OT-comparative rows; in the latter case, HG-comparative rows are derived from both OT-comparative rows and weight vectors, as the weight vector is used in (81) to determine the *undominated* loser-preferrers. Thus, the notation "$\overline{\mathbf{a}}$" for the derived HG-comparative row in (81) is not optimal now, as it hides the dependence on the current weight vector; but I will stick to this notation, for coherence with the rest of the paper. Let $\overline{\mathbf{a}}^t$ be the HG-comparative row derived according to (81) from the pair $(\mathbf{a}^t, \boldsymbol{\theta}^t)$ of the current comparative row and current weight vector in the run (80).

(81)
$$\big(\mathbf{a}, \boldsymbol{\theta}\big) \downarrow \quad \overline{\mathbf{a}} = \big[\overline{a}_1, \ldots, \overline{a}_n\big] \quad \text{where } \overline{a}_k \doteq \begin{cases} -w & \text{if } C_k \text{ is an undominated loser-preferrer} \\ 0 & \text{if } a_k = \text{E} \\ \ell & \text{if } C_k \text{ is a winner-preferrer} \end{cases}$$

By reasoning just as I did above in Subsection 4.4 for the example in (70), it turns out that update of the current weight vector $\boldsymbol{\theta}^t$ according to the new promotion/demotion OT update rule (79) in response to the OT-comparative row $\mathbf{a}^t$ is equivalent to update according to the Perceptron HG-update rule (53) in response to the HG-comparative row $\overline{\mathbf{a}}^t$ derived through (81) from the pair $(\mathbf{a}^t, \boldsymbol{\theta}^t)$ of the current OT comparative row and the current weight vector. The situation (80) thus entails (82).

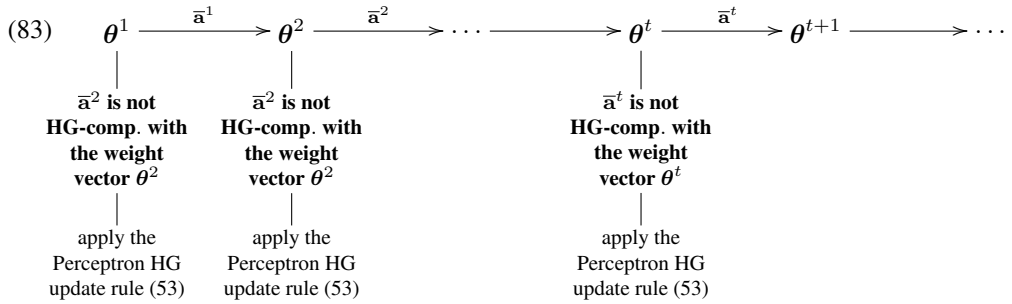
(82)


The mapping (69) used in Subsection 4.4 had the following crucial property: if the OT-comparative row you start with is not OT-compatible with some ranking derived from a weight vector, then the corresponding derived HG-comparative row is not HG-compatible with that weight vector, as guaranteed by claim 5. Claim 11 ensures that an analogous property holds for the mapping (81) used here. The proof of claim 11 is presented in Appendix A.5, and it is just a very small variant of the proof of claim 11 presented in Appendix A.2.

Claim 11 *Consider an OT-comparative row* $\mathbf{a}$ *and a weight vector* $\boldsymbol{\theta}$ *and let* $\overline{\mathbf{a}}$ *be the HG-comparative row derived from them according to (82). If the OT-comparative row* $\mathbf{a}$ *is not OT-compatible with some ranking derived from the weight vector* $\boldsymbol{\theta}$*, then the corresponding derived HG-comparative row* $\overline{\mathbf{a}}$ *is not HG-compatible with* $\boldsymbol{\theta}$ *either.* ■

The latter claim 11 now plays a crucial role. The current OT-comparative row $\mathbf{a}^t$ fed to the algorithm in (82) is not OT-compatible with some ranking derived from the current weight vector $\boldsymbol{\theta}^t$. Claim 11 thus ensures that the derived HG-comparative row $\overline{\mathbf{a}}^t$ is not HG-compatible with the current weight vector $\boldsymbol{\theta}^t$ either. The situation (71) thus entails (83).

(83)



Collect together all the HG-comparative rows $\overline{\mathbf{a}}_1, \overline{\mathbf{a}}_2, \ldots, \overline{\mathbf{a}}_t, \ldots$ fed to the HG online algorithm in the run (83). Note that there is only a finite number of them. Stack these finitely many HG-comparative rows one on top of the other into an HG-comparative tableau $\overline{\mathbf{A}}$. This tableau is derived from the input OT-comparative tableau $\mathbf{A}$ according to definition (41). Since OT-compatibility entails HG-compatibility of any derived tableau by claim 4, then this HG-comparative tableau $\overline{\mathbf{A}}$ is HG-compatible. The diagram in (83) thus contradicts claim 6, that ensures convergence for the Perceptron HG update rule (53). We have thus proved the following claim 12.

Claim 12 *The OT online algorithm (56) with the new general promotion/demotion OT update rule (79) converges, provided that the input comparative tableau is OT-compatible.* ■

4.7 A more abstract look

In this section, I would like to look at the reasoning developed so far from a more abstract perspective, that will reveal the close connection to the general algorithmic scheme (47). In subsection 4.2, I have worked on the *right-hand side* of the algorithmic scheme (47): using the notion (39) of a ranking *derived* from a weight vector, I have described OT online algorithms in terms of weight vectors (with no restrictions on the sign of the weights) rather than in terms of rankings, as in (56). And I have noted that this allows for OT update rules to be defined in terms of weight vectors rather than in terms of rankings, as in (57). In Subsections 4.4 and 4.6, I have then turned to the *left-hand side* of the algorithmic scheme (47). Let me review what I have done. Consider a mapping of the form (84a) that takes an OT-comparative row $\mathbf{a}$ and a weight vector $\boldsymbol{\theta}$ and constructs out of them an HG-comparative row $\overline{\mathbf{a}}$. Once OT update rules are restated as in (57) in terms of weight vectors, an OT update rule (57) and an HG update rule (51) only differ because the former takes an OT-comparative row $\mathbf{a}$ while the latter takes an HG-comparative row $\overline{\mathbf{a}}$. Using a mapping (84a) from OT- into HG-comparative rows, we can thus "lift" or "translate" any HG update rule into the corresponding *derived* OT update rule according to the general scheme (84b): given a pair $(\mathbf{a}, \boldsymbol{\theta}^{\mathrm{old}})$ of the current OT-comparative row $\mathbf{a}$ and the current weight vector $\boldsymbol{\theta}^{\mathrm{old}}$, we consider the corresponding HG-comparative row $\overline{\mathbf{a}}$ through (84a) and apply the HG update rule to the pair $(\overline{\mathbf{a}}, \boldsymbol{\theta}^{\mathrm{old}})$. The definition of derived OT update rules in terms of the scheme (84b) illustrates the general algorithmic scheme (47).

(84) a. $(\mathbf{a}, \boldsymbol{\theta})$ $\Rightarrow$ $\overline{\mathbf{a}}$ b. $\underbrace{(\mathbf{a}, \boldsymbol{\theta}^{\text{old}})}$ $\Rightarrow$ $(\overline{\mathbf{a}}, \boldsymbol{\theta}^{\text{old}})$ $\xrightarrow{\text{HG-update rule}}$ $\boldsymbol{\theta}^{\text{new}}$, $\underbrace{(\mathbf{a}, \boldsymbol{\theta}^{\text{old}})}$ $\xrightarrow{\text{OT-update rule}}$ $\boldsymbol{\theta}^{\text{new}}$

Let me collect together various examples considered in Subsections 4.4 and 4.6 from this perspective. I need to start from a specific mapping (84a) from OT-comparative rows and weight vectors into derived HG-comparative rows. Consider the one in (85a): an L is replaced with -1, an E with 0, and a W with $+1$. In this case, the derived HG-comparative row does not actually depend on the weight vector. Given the Perceptron HG update rule (53), I can then construct the corresponding derived OT update rule by applying the general scheme (84b) with the mapping (85a). What I get is (85b), namely Boermsa's OT update rule (61).

(85) a. $\overline{a}_k = \begin{cases} 1 & \text{if } a_k = \text{W} \\ -1 & \text{if } a_k = \text{L} \\ 0 & \text{if } a_k = \text{E} \end{cases}$ b. $\theta_k^{\text{new}} = \begin{cases} \theta_k^{\text{old}} + 1 & \text{if } C_k \text{ is winner-preferrer} \\ \theta_k^{\text{old}} - 1 & \text{if } C_k \text{ is loser-preferrer} \\ \theta_k^{\text{old}} & \text{otherwise} \end{cases}$

Let me consider another example. Let me assume that the input OT-comparative rows have a unique L per row. Again, I need to start from a specific mapping (84a) from OT-comparative rows and weight vectors into derived HG-comparative rows. Consider the one in (86a): a W is replaced with $+1$, an E with 0, and the unique L with $-w$, where w is the total number of W's. Again also in this case, the derived HG-comparative row does not actually depend on the weight vector. Given the Perceptron HG update rule (53), I can then construct the corresponding derived OT update rule by applying the general scheme (84b) with the mapping (86a). What I get is (86b), namely the new cautious promotion/demotion OT update rule (66) devised in Subsection 4.4 for input rows with a unique L per row.

(86) a. $\overline{a}_k = \begin{cases} 1 & \text{if } a_k = \text{W} \\ -w & \text{if } a_k = \text{L} \\ 0 & \text{if } a_k = \text{E} \end{cases}$ b. $\theta_k^{\text{new}} = \begin{cases} \theta_k^{\text{old}} + 1 & \text{if } C_k \text{ is winner-preferrer} \\ \theta_k^{\text{old}} - w & \text{if } C_k \text{ is the loser-preferrer} \\ \theta_k^{\text{old}} & \text{otherwise} \end{cases}$

Let me consider one more example. Again, I need to start from a specific mapping (84a) from OT-comparative rows and weight vectors into derived HG-comparative rows. Consider the one in (87a): the entries corresponding to undominated loser-preferrers are replaced with $-w$, where w is the total number of winner-preferrers; and the entries corresponding to winner-preferrers are replaced with the total number ℓ of undominated loser-preferrers. Note that this mapping (87a) depends on the OT-comparative row as well as on the weight vector, as it involves the number of currently *undominated* loser-preferrers. Given the Perceptron HG update rule (53), I can then construct the corresponding derived OT update rule by applying the general scheme (84b) with the mapping (87a). What I get is (87b), namely the new general promotion/demotion OT update rule (79) devised in Subsection 4.6.

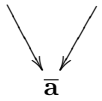
(87) a. $\overline{a}_k = \begin{cases} \ell & \text{if } C_k \text{ is winner-preferrer} \\ -w & \text{if } C_k \text{ is an undominated loser-preferrer} \\ 0 & \text{otherwise} \end{cases}$ b. $\theta_k^{\text{new}} = \begin{cases} \theta_k^{\text{old}} + \ell & \text{if } C_k \text{ is winner-preferrer} \\ \theta_k^{\text{old}} - w & \text{if } C_k \text{ is an undominated loser-preferrer} \\ \theta_k^{\text{old}} & \text{otherwise} \end{cases}$

I have considered in (85a), (86a) and (87a) examples of a mapping of the form (84a) from OT-comparative rows and weight vectors into HG-comparative rows. These examples share some important properties. Let me make them explicit. All three examples satisfy condition (88): W's (L's) of the OT-comparative row are replaced with positive (non-positive) numbers. These numbers can depend on the weight vector $\boldsymbol{\theta}$.

$$(88)\quad \overline{a}_k \begin{cases} > 0 & \text{if } a_k = \text{W} \\ = 0 & \text{if } a_k = \text{E} \\ \leq 0 & \text{if } a_k = \text{L} \end{cases}$$

Yet, the dependence on the weight vector is very limited: in the case of the first two examples (85a) and (86a), there is actually no dependence on the current weight vector at all; and in the case of the third example (87a), the dependence is limited in the sense that the mapping does not distinguish between two weight vectors that have exactly the same derived rankings, as schematized in (89). In other words, the mapping is only sensitive to the properties of the weight vector that are relevant from the perspective of OT (and not other irrelevant properties, such as the absolute size of the weights).

(89) $(\mathbf{a}, \boldsymbol{\theta})$ $(\mathbf{a}, \boldsymbol{\theta}')$ where $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ have exactly the same set of derived rankings



We hope that, if the HG update rule we start from is convergent for the HG online algorithm, then the corresponding OT update rule derived through (84b) is convergent for the OT online algorithm. Whether that is indeed the case, depends of course on the specific mapping (84a) from OT- into derived HG-comparative rows used to implement the scheme (84b). For example, if we adopt the mapping from OT- into derived HG-comparative rows in (85a), then the scheme does not preserve convergence: we start from the convergent HG Perceptron update rule (53) and we obtain Boersma's non-convergent OT update rule (61). These considerations motivate the following research goal: to characterize mappings (84a) from OT- into derived HG-comparative rows such that the corresponding scheme (84b) *preserves convergence*, namely it turns convergent HG update rules into convergent OT update rules. To this end, suppose that the mapping (84a) satisfies the crucial condition (90), for any OT-comparative row $\mathbf{a}$ and any weight vector $\boldsymbol{\theta}$.

(90) If the OT-comparative row $\mathbf{a}$ is *not* OT-compatible with a ranking derived from the weight vector $\boldsymbol{\theta}$, then the corresponding HG-comparative row $\overline{\mathbf{a}}$ derived through (84a) is *not* HG-compatible with $\boldsymbol{\theta}$.

As seen above, claims 5 and 11 guarantee that the two mappings (86a) and (87a) satisfy the crucial condition (90). The mapping (85a), that corresponds to Boersma's non-convergent update rule (61), instead does not satisfy the crucial condition (90). Here is a counterexample. Consider the OT-comparative row $\mathbf{a}$ in (91a); the corresponding HG-comparative row derived through (85a) is (91c). Consider the weight vector $\boldsymbol{\theta}$ in (91d) and one if its refinements (91b). The OT comparative row (91a) is not OT-compatible with the ranking (91b), despite the fact that the HG-comparative row (91c) is HG-compatible with the weight vector (91d).

(91) a. $\mathbf{a} = \begin{bmatrix} C_1 & C_2 & C_3 \\ \text{W} & \text{W} & \text{L} \end{bmatrix}$ b. $C_3 \gg C_2 \gg C_1$

c. $\overline{\mathbf{a}} = \begin{bmatrix} C_1 & C_2 & C_3 \\ 1 & 1 & -1 \end{bmatrix}$ d. $\boldsymbol{\theta} = \begin{bmatrix} C_1 & C_2 & C_3 \\ 2 & 2 & 3 \end{bmatrix}$

The reasoning presented in Subsections 4.4 and repeated in Subsection 4.6 guarantees that the crucial property (90) allows convergent OT update rules to be derived from convergent HG update rules, as stated in claim 13. Let me summarize here in three core steps the reasoning behind claim 13. *First*, if the current OT-comparative row $\mathbf{a}^t$ is not OT-compatible with some ranking derived from the current weight vector $\boldsymbol{\theta}^t$, then the corresponding HG-comparative row $\overline{\mathbf{a}}^t$ derived through (84a) is not HG-compatible with the current weight vector either, by the crucial assumption (90). *Second*, update of the current weight vector $\boldsymbol{\theta}^t$ triggered by the current OT-comparative row $\mathbf{a}^t$ according to the derived OT update rule is equivalent to update of the current weight vector $\boldsymbol{\theta}^t$ triggered by the corresponding derived HG-comparative row $\overline{\mathbf{a}}^t$ according to the original HG update rule. *Third*, the HG-comparative rows derived from the input OT-comparative rows through (84a) are only a finite number, because of the loose dependence (89) on the weight vector. Stack these finitely many HG-comparative rows one on top of the other into an HG-comparative tableau $\overline{\mathbf{A}}$. By assumption (88),

$\overline{\mathbf{A}}$ is derived from the input tableau $\mathbf{A}$ and claim 4 thus ensures that $\overline{\mathbf{A}}$ is HG-compatible, as it is derived from an OT-compatible tableau.

Claim 13 *Consider a mapping (84a) from OT-comparative rows and weight vectors into derived HG-comparative rows. Assume that it satisfies conditions (88), (89) and (90). Then, the corresponding scheme (84b) from HG update rules into derived OT update rules preserves convergence, namely it turns* any *convergent HG update rule into a convergent OT update rule.* ■

A Appendices

A.1 Proof of claim 1

In Section 2, I noted that what is currently known in the literature concerning the relationship between the two frameworks of OT and HG is that OT-compatibility entails HG-compatibility, as stated in claim 1 repeated below. In this subsection, I present the classical proof of this claim, from Prince and Smolensky (2004) and Keller (2000, 2005).

Claim 1 *If a set $\mathcal{D}$ of underlying/winner/loser form triplets is OT-compatible, then it is also HG-compatible. More precisely, let $\gg$ be a ranking OT-compatible with $\mathcal{D}$. Without loss of generality, assume that it is (92a), with C_n ranked at the top, C_{n-1} below it and so on, until the bottom ranked C_1. Then, the weight vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ defined in (92b) is HG-compatible with $\mathcal{D}$*

(92) *a.*
$$\begin{array}{c} C_n \\ | \\ C_{n-1} \\ | \\ \vdots \\ | \\ C_1 \end{array}$$

b.
$$\begin{aligned} \theta_n &= (\delta+1)^n \\ \theta_{n-1} &= (\delta+1)^{n-1} \\ &\vdots \\ \theta_1 &= (\delta+1) \end{aligned}$$

where δ is the largest constraint difference (ignoring sign) over all constraints and all data triplets in the data set $\mathcal{D}$. ■

Proof. Let $\overline{\mathbf{A}} = \overline{\mathbf{A}}(\mathcal{D})$ be the HG-comparative tableau of constraint differences corresponding to the set of data triplets $\mathcal{D}$; let $\mathbf{A} = \mathbf{A}(\mathcal{D})$ be the corresponding OT-comparative tableau. Consider an arbitrary row $\overline{\mathbf{a}} = [\overline{a}_1, \ldots, \overline{a}_n]$ of the HG-comparative tableau $\overline{\mathbf{A}}$. Let $\mathbf{a} = [a_1, \ldots, a_n]$ be the corresponding OT-comparative row of $\mathbf{A}$. Since the ranking (92a) satisfies the OT-compatibility condition (25) with this row $\mathbf{a}$, then there is some $k \in \{n, n-1, \ldots, 1\}$ such that the top ranked constraints $C_n, C_{n-1}, \ldots, C_{k+1}$ are even for the OT-comparative row $\mathbf{a}$ and constraint C_k is winner-preferrer, as stated in (93a). This means that the corresponding HG-comparative row of constraint differences satisfies (93b).

(93) a. $a_n = a_{n-1} = \ldots = a_{k+1} = \text{E},\ a_k = \text{W}.$

b. $\overline{a}_n = \overline{a}_{n-1} = \ldots = \overline{a}_{k+1} = 0,\ \overline{a}_k > 0.$

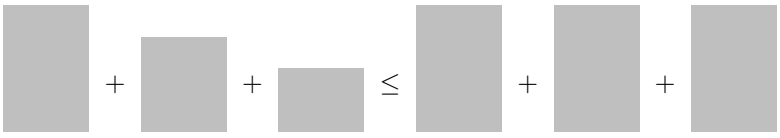
To simplify notation, let $B = \delta + 1$. The chain of inequalities in (94) shows that the weight vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$ in (92) does indeed satisfy the HG-compatibility condition $\sum_{i=1}^{n} \theta_i \overline{a}_i > 0$ in (20).

$$
\begin{aligned}
(94)\quad \sum_{i=1}^{n}\theta_i\overline{a}_i &\overset{(a)}{=} \sum_{i=1}^{k-1}\theta_i\overline{a}_i + \theta_k\overline{a}_k + \sum_{i=k+1}^{n}\theta_i\overline{a}_i \\
&\overset{(b)}{=} \sum_{i=1}^{k-1}\theta_i\overline{a}_i + \theta_k\overline{a}_k \\
&\overset{(c)}{\geq} \sum_{i=1}^{k-1}\theta_i\overline{a}_i + \theta_k \\
&\overset{(d)}{\geq} -\sum_{i=1}^{k-1}\theta_i(B-1) + \theta_k \\
&\overset{(e)}{=} -\sum_{i=1}^{k-1}B^i(B-1) + B^k \\
&= -\sum_{i=1}^{k-1}B^{i+1} + \sum_{i=1}^{k-1}B^i + B^k \\
&= -\sum_{i=2}^{k}B^i + \sum_{i=1}^{k-1}B^i + B^k \\
&= -B^k + B + B^k \\
&> 0
\end{aligned}
$$

In (94), I have reasoned as follows: in step (a), I have split up the set $\{1,\ldots,n\}$ that the index i runs over into the three subsets $\{1,\ldots,k-1\}$, $\{k\}$ and $\{k+1,\ldots,n\}$; in step (b), I have used the fact that $\overline{a}_n = \overline{a}_{n-1} = \ldots = \overline{a}_{k+1} = 0$ by (93b); in step (c), I have lower-bounded by replacing $\overline{a}_k$ with 1, since the fact that $\overline{a}_k > 0$ by (93b) entails that $\overline{a}_k \geq 1$, as constraint differences are integers; in step (d), I have lower-bounded by replacing $\overline{a}_1,\ldots,\overline{a}_{k-1}$ with $-(B-1)$, since the absolute value of the entries in the HG-comparative row is upper bounded by $\delta = B-1$; in step (e), I have used the definition (92) of the weight vector $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_n)$, restated in terms of $B = \delta + 1$; the remaining steps are simple algebraic manipulations. □

A.2 Proof of claims 2 and 5

In this Subsection, I present a proof of the main claim 2, repeated below. The proof is actually just a straightforward generalization of the reasoning illustrated in subsection 3.1 with the three examples (35)-(37). It rests on the trivial observation (95). Suppose we have a certain number of weights, say three. The sum of these three weights is always upper bounded by taking three times the largest among them.

(95) 

Claim 2 *Given an OT-comparative tableau* $\mathbf{A}$*, consider the corresponding HG-comparative tableau* $\overline{\mathbf{A}}$ *derived from* $\mathbf{A}$ *row-by-row as in (38), repeated in (96). Here,* w *is the total number of* W*'s in the OT-comparative row* $\mathbf{a}$*.*

(96) $\mathbf{a} = \big[a_1,\ldots,a_n\big] \longrightarrow \overline{\mathbf{a}} = \big[\overline{a}_1,\ldots,\overline{a}_n\big]$ *such that* $\overline{a}_k \doteq \begin{cases} -w & \text{if } a_k = \text{L} \\ 0 & \text{if } a_k = \text{E} \\ 1 & \text{if } a_k = \text{W} \end{cases}$

If $\overline{\mathbf{A}}$ *is HG-compatible, then* $\mathbf{A}$ *is OT-compatible. Furthermore, if* $\boldsymbol{\theta}$ *is a solution of the instance WP*$(\overline{\mathbf{A}})$ *of the Weighting problem, then any ranking derived from* $\boldsymbol{\theta}$ *according to (39) is a solution of the instance RP*$(\mathbf{A})$ *of the Ranking problem.* ■

Proof. Let me show that, if a weight vector $\boldsymbol{\theta}$ is HG-compatible with the HG-comparative row $\overline{\mathbf{a}}$ in (96), then any of its derived rankings is OT-compatible with the original OT-comparative row $\mathbf{a}$. Let W and L be the sets of constraints that have a W and an L in the OT-comparative row $\mathbf{a}$, respectively. Let w be cardinality of the set W, namely the total number of constraints that have an W in row $\mathbf{a}$. The following chain of inequalities (97) holds for every loser-preferring constraint $k \in L$. Here, I have reasoned as follows: in step (97a), I have used the hypothesis that the weight vector $\boldsymbol{\theta}$ is HG-compatible with $\overline{\mathbf{a}}$ and thus satisfies condition (20); in step (97b), I have split up the set $\{1, \ldots, n\}$ that h runs over into the three sets W, L and their complement; in step (97c), I have noted that $\overline{a}_h = 1$ for every $h \in W$, that $\overline{a}_h = -w$ for every $h \in L$ and that $\overline{a}_h = 0$ for every $h \notin W \cup L$, by the definition (96) of the derived HG-comparative row $\overline{\mathbf{a}}$; in step (97d), I have used (95) to upper bound the sum $\sum_{h \in W} \theta_h$ with its biggest term $\max_{h \in W} \theta_h$ multiplied by the total number w of terms; in step (97e), I have used the hypothesis that all the components of $\boldsymbol{\theta}$ are nonnegative and thus $\sum_{h \in L} \theta_h \geq \theta_k$ provided that $k \in L$.

$$
\begin{aligned}
(97) \qquad 0 &\overset{(a)}{<} \sum_{h=1}^{n} \theta_h \overline{a}_h \\
&\overset{(b)}{=} \sum_{h \in W} \theta_h \overline{a}_h + \sum_{h \in L} \theta_h \overline{a}_h + \sum_{h \notin W \cup L} \theta_h \overline{a}_h \\
&\overset{(c)}{=} \sum_{h \in W} \theta_h - w \sum_{h \in L} \theta_h \\
&\overset{(d)}{\leq} w \max_{h \in W} \theta_h - w \sum_{h \in L} \theta_h \\
&\overset{(e)}{\leq} w \max_{h \in W} \theta_h - w \theta_k
\end{aligned}
$$

By reordering the chain of inequalities in (97), I conclude that the strict inequality $\max_{h \in W} \theta_h > \theta_k$ holds for any loser-preferrer constraint $k \in L$. This inequality guarantees that the weight vector $\boldsymbol{\theta}$ has the following property: the largest weight among winner-preferers is strictly larger than the weights of loser-preferrers. The definition (39) of derived rankings thus guarantees that any ranking derived from $\boldsymbol{\theta}$ ranks a winner-preferrer above all loser-preferrers and is thus OT-compatible with the original OT-comparative row $\mathbf{a}$. □

Note that the assumption (11) that the weights $\theta_1, \ldots, \theta_n$ are non-negative was used only once in the preceding proof, namely in step (97e), in order to upper-bound the sum $\sum_{h \in L} \theta_h$ of the weights of loser-preferrers with one of those weights. If the assumption (11) that the weights be non-negative is dropped, then the upper-bound in (97e) does not hold any more. Yet the bound trivially holds, even without the non-negativity restriction, provided that row $\mathbf{a}$ has a unique loser-preferring constraint C_k, as in this case $L = \{k\}$ and thus $\sum_{h \in L} \theta_h = \theta_k$. This is why claim 5 holds.

A.3 Proof of claim 6

In this Subsection, I recall for completeness the proof of the classical Perceptron convergence claim 6, repeated below; see for instance Cristianini and Shawe-Taylor (2000, Theorem 2.3), and Cesa-Bianchi and Lugosi (2006, Chp. 12) for a broader perspective. In this subsection, I use standard notation from Linear Algebra: $\langle \cdot, \cdot \rangle$ is the *Euclidean scalar product*, defined by $\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{i=1}^{n} v_i w_i$ for any pair of vectors $\mathbf{v} = (v_1, \ldots, v_n)$ and $\mathbf{w} = (w_1, \ldots, w_n)$; $\| \cdot \|$ is the *Euclidean norm*, defined by $\|\mathbf{v}\|^2 = \langle \mathbf{v}, \mathbf{v} \rangle = \sum_{i=1}^{n} v_i^2$; they are connected by *Cauchy-Schwartz inequality* $\langle \mathbf{v}, \mathbf{w} \rangle \leq \|\mathbf{v}\| \|\mathbf{w}\|$. The HG-compatibility condition (20) between a weight vector $\boldsymbol{\theta}$ and an HG-comparative row $\overline{\mathbf{a}}$ can be expressed in terms of scalar product as $\langle \boldsymbol{\theta}, \overline{\mathbf{a}} \rangle > 0$.

Claim 6 *The HG online algorithm (50) with the HG Perceptron update rule (53) converges, provided that the input comparative tableau is HG-compatible.* ■

Proof. Without loss of generality, assume that the initial weight vector is the null vector. Let $\boldsymbol{\theta}^{t+1}$ be the weight vector obtained at time t by updating the current weight vector $\boldsymbol{\theta}^t$ in response to the current HG-comparative row $\mathbf{a}^t$ according to the Perceptron update rule (53). As current rows that do not trigger an update can be discarded, I can assume without loss of generality that an update is performed at each time t. The proof has three parts. The *first* part of the proof estimates the norm of the current weight vector entertained by the algorithm. To start, note that the norm of the updated weight vector $\boldsymbol{\theta}^{t+1}$ can be bound as in (98) in terms of the norm of the current weight vector $\boldsymbol{\theta}^t$. Here, I have reasoned as follows: in step (a), I have used the definition (53) of the Perceptron HG update rule; in step (b), I have used the identity $\|\mathbf{v}+\mathbf{w}\|^2 = \|\mathbf{v}\|^2 + \|\mathbf{w}\|^2 + 2\langle \mathbf{v}, \mathbf{w}\rangle$; in step (c), I have upper bounded by dropping the term $\langle \boldsymbol{\theta}^t, \overline{\mathbf{a}}^t\rangle$, which is negative by the hypothesis that the current weight vector $\boldsymbol{\theta}^t$ is not HG-compatible with the current HG-comparative row $\overline{\mathbf{a}}^t$; in step (d), I have upper bound the norm of the current HG-comparative row $\overline{\mathbf{a}}^t$ with the largest norm over all input HG-comparative rows, called R.

$$
\begin{aligned}
(98) \quad \|\boldsymbol{\theta}^{t+1}\|^2 &\overset{(a)}{=} \|\boldsymbol{\theta}^t + \overline{\mathbf{a}}^t\|^2 \\
&\overset{(b)}{=} \|\boldsymbol{\theta}^t\|^2 + \|\overline{\mathbf{a}}^t\|^2 + 2\langle \boldsymbol{\theta}^t, \overline{\mathbf{a}}^t\rangle \\
&\overset{(c)}{\leq} \|\boldsymbol{\theta}^t\|^2 + \|\overline{\mathbf{a}}^t\|^2 \\
&\overset{(d)}{\leq} \|\boldsymbol{\theta}^t\|^2 + R^2 \\
&\qquad \text{where } R = \max\left\{ \|\overline{\mathbf{a}}\| \;\middle|\; \overline{\mathbf{a}} \text{ is a row of the input HG-comparatve tableau} \right\}
\end{aligned}
$$

Since the inequality (98) holds at every time t and since the initial weight vector has null norm, then we obtain the inequality (99), which concludes the first part of the proof, showing that the norm of the current weight vector grows with time t slower than $\sqrt{t}$.

$$
(99) \quad \|\boldsymbol{\theta}^t\|^2 \leq tR^2
$$

Consider now a weight vector $\boldsymbol{\theta}$ HG-compatible with the input HG-comparative tableau, that exists because of the hypothesis that the latter is HG-compatible. The *second* part of the proof estimates the scalar product between the latter weight vector and the current weight vector entertained by the algorithm. To start, note that the scalar product between the weight vector $\boldsymbol{\theta}$ and the updated weight vector $\boldsymbol{\theta}^{t+1}$ can be bound as in (100) in terms of the scalar product between that same weight vector $\boldsymbol{\theta}$ and the current weight vector $\boldsymbol{\theta}^t$. Here, I have reasoned as follows: in step (a), I have used the definition (53) of the Perceptron HG update rule; in step (b), I have used the linearity of the scalar product; in step (c), I have lower bounded the quantity $\langle \boldsymbol{\theta}, \overline{\mathbf{a}}^t\rangle / \|\boldsymbol{\theta}\|$ with the smallest such quantity over all input HG-comparative rows, called $\mu(\boldsymbol{\theta})$.

$$
\begin{aligned}
(100) \quad \frac{\langle \boldsymbol{\theta}, \boldsymbol{\theta}^{t+1}\rangle}{\|\boldsymbol{\theta}\|} &\overset{(a)}{=} \frac{\langle \boldsymbol{\theta}, \boldsymbol{\theta}^t + \overline{\mathbf{a}}^t\rangle}{\|\boldsymbol{\theta}\|} \\
&\overset{(b)}{=} \frac{\langle \boldsymbol{\theta}, \boldsymbol{\theta}^t\rangle}{\|\boldsymbol{\theta}\|} + \frac{\langle \boldsymbol{\theta}, \overline{\mathbf{a}}^t\rangle}{\|\boldsymbol{\theta}\|} \\
&\overset{(c)}{\geq} \frac{\langle \boldsymbol{\theta}, \boldsymbol{\theta}^t\rangle}{\|\boldsymbol{\theta}\|} + \mu(\boldsymbol{\theta}) \\
&\qquad \text{where } \mu(\boldsymbol{\theta}) = \min\left\{ \tfrac{\langle \boldsymbol{\theta}, \overline{\mathbf{a}}\rangle}{\|\boldsymbol{\theta}\|} \;\middle|\; \overline{\mathbf{a}} \text{ is a row of the input HG-comparatve tableau} \right\}
\end{aligned}
$$

Since the inequality (100) holds at every time t and since the initial weight vector is null, then we obtain the inequality (101), which concludes the second part of the proof, showing that the scalar product between the weight vector $\boldsymbol{\theta}$ and the current weight vector $\boldsymbol{\theta}^t$ grows faster than t.

$$
(101) \quad \frac{\langle \boldsymbol{\theta}, \boldsymbol{\theta}^t\rangle}{\|\boldsymbol{\theta}\|} \geq t\mu.
$$

The *third* part of the proof connects the two inequalities (99) and (101) as in (102). Here, I have reasoned as follows: in step (a), I have used (101); in step (b), I have used the Cauchy-Schwartz inequality; in step (c), I have used (99).

$$
\begin{array}{rcl}
(102) \quad \big(t\mu\big)^2 & \overset{(a)}{\leq} & \left(\dfrac{\langle \boldsymbol{\theta}, \boldsymbol{\theta}^t \rangle}{\|\boldsymbol{\theta}\|}\right)^2 \\
& \overset{(b)}{\leq} & \dfrac{\|\boldsymbol{\theta}\|^2 \|\boldsymbol{\theta}^t\|^2}{\|\boldsymbol{\theta}\|^2} \\
& = & \|\boldsymbol{\theta}^t\|^2 \\
& \overset{(c)}{\leq} & tR^2
\end{array}
$$

The chain of inequalities (102) entails in particular that $t \leq R^2/\mu^2$, namely that the number of updates t is finite. □

The proof just presented actually shows that the total number T of updates performed by the HG online algorithm can be bound as in (103), in terms of the two quantities R and $\mu(\boldsymbol{\theta})$ defined in (98) and (100).

$$
(103) \quad T \leq \left(\frac{R}{\mu(\boldsymbol{\theta})}\right)^2
$$

The bound can be optimized by considering the weight vector $\hat{\boldsymbol{\theta}}$ that maximizes $\mu(\boldsymbol{\theta})$. The corresponding quantity $\mu = \mu(\hat{\boldsymbol{\theta}})$ is called the *margin* of the input HG-comparative tableau.

A.4 Proof of claims 9 and 10

In this Subsection, I present a proof of claims 9 and 10 repeated below, that offer bounds on the worst-case number of updates performed by the OT online algorithm with the new cautious promotion-demotion OT update rule (66) on *diagonal* and on *Pater's comparative tableaux*. The core idea of the proof is as follows. As noted in Subsection 4.4, this new update rule (66) is derived from the HG Perceptron update rule (53). Claim 8 thus ensures that bounds on the worst-case number of updates performed by the OT online algorithm can be obtained from the bounds on the worst-case number of updates performed by the Perceptron HG online algorithm, such as the bound (103) at the end of Appendix A.3. In this Subsection, I use standard notation form Linear Algebra: $\langle \cdot, \cdot \rangle$ is the Euclidean scalar product and $\| \cdot \|$ is the Euclidean norm.

Claim 9 *The worst-case number of updates performed by the OT online algorithm (56) with the new cautious promotion-demotion OT update rule (66) run on the diagonal comparative tableau of order n starting from the null initial vector is bound by $n(n^2-1)/6$.* ■

Proof. Consider the mapping from an OT-comparative row $\mathbf{a} = [a_1, \ldots, a_n]$ into an HG-comparative row $\overline{\mathbf{a}} = [\overline{a}_1, \ldots, \overline{a}_n]$ defined in (38) and (69), repeated below.

$$
(104) \quad \overline{a}_k = \begin{cases} +1 & \text{if } a_k = \text{W} \\ 0 & \text{if } a_k = \text{E} \\ -w & \text{if } a_k = \text{L} \end{cases} \qquad \text{where } w \text{ is the total number of winner-preferrers in row } \mathbf{a}
$$

Let me denote by $\overline{\mathbf{A}}_n$ the HG-comparative tableau derived through the mapping (104) from the diagonal comparative tableau $\mathbf{A}_n$ of order n. To illustrate, I provide in (105) the derived HG-comparative tableaux corresponding to the three diagonal tableaux in (74).

$$
(105) \quad \overline{\mathbf{A}}_4 = \begin{bmatrix} 1 & -1 & & \\ & 1 & -1 & \\ & & 1 & -1 \end{bmatrix} \quad \overline{\mathbf{A}}_5 = \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & 1 & -1 & \\ & & & 1 & -1 \end{bmatrix} \quad \overline{\mathbf{A}}_6 = \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & 1 & -1 & & \\ & & & 1 & -1 & \\ & & & & 1 & -1 \end{bmatrix}
$$

As noted in Subsection 4.4, the new cautious promotion/demotion update rule (66) is derived through mapping (104) from the HG Perceptron update rule (53). Claim 8 thus ensures that the worst case

number of updates $T_{\text{OT}}(n)$ performed by the OT online algorithm with this new cautious promotion/demotion OT update rule on the diagonal comparative tableau $\mathbf{A}_n$ is at most as large as the worst case number of updates $T_{\text{HG}}(n)$ performed by the HG online algorithm with the HG Perceptron update rule (53) on the derived diagonal HG-comparative tableau $\overline{\mathbf{A}}_n$, as in (106a). As noted at the end of Appendix A.3, the latter worst-case number $T_{\text{HG}}(n)$ of HG updates can in turn be bound as in (106b).

$$T_{\text{OT}}(n) \overset{\text{(a)}}{\leq} T_{\text{HG}}(n) \overset{\text{(b)}}{\leq} \frac{R^2}{\mu^2} \qquad \begin{array}{l} \text{where } R = R(\overline{\mathbf{A}}_n) = \text{maximum norm of the rows of } \overline{\mathbf{A}}_n \\ \qquad\quad \mu = \mu(\overline{\mathbf{A}}_n) = \text{margin of the rows of } \overline{\mathbf{A}}_n \end{array} \tag{106}$$

The squared norm of any row of $\overline{\mathbf{A}}_n$ is 2. In order to conclude the proof, I thus only need to lower bound the squared inverse $1/\mu^2$ of the margin. Vapnik (1998, Theorem 10.2) ensures that $1/\mu^2$ coincides with the (unique) solution of the quadratic optimization problem (107) in the decision variable $\boldsymbol{\theta} \in \mathbb{R}^n$.

$$\begin{array}{lll} \text{minimize:} & \|\boldsymbol{\theta}\|^2 & \\ \text{subject to:} & \langle \boldsymbol{\theta}, \overline{\mathbf{a}} \rangle \geq 1 & \text{for every row } \overline{\mathbf{a}} \text{ of the derived diagonal tableau } \overline{\mathbf{A}}_n \end{array} \tag{107}$$

A row $\overline{\mathbf{a}}$ of the derived HG-comparative diagonal tableau $\overline{\mathbf{A}}_n$ is called a *support vector* iff the condition $\langle \boldsymbol{\theta}, \overline{\mathbf{a}} \rangle \geq 1$ in the definition of the feasible set in (107) holds tight at optimality, namely $\langle \boldsymbol{\theta}^*, \overline{\mathbf{a}} \rangle = 1$, where $\boldsymbol{\theta}^*$ is the unique solution of the optimization problem (107). In the special case of diagonal comparative tableaux, all rows are support vectors, so that the optimization problem (107) is equivalent to (108).[10]

$$\begin{array}{lll} \text{minimize:} & \|\boldsymbol{\theta}\|^2 & \\ \text{subject to:} & \langle \boldsymbol{\theta}, \overline{\mathbf{a}} \rangle = 1 & \text{for every row } \overline{\mathbf{a}} \text{ of the derived diagonal tableau } \overline{\mathbf{A}}_n \end{array} \tag{108}$$

Consider the diagonal comparative tableau of order $n = 5$, repeated in (109) together with the corresponding derived diagonal HG-comparative tableau. For convenience, I have numbered the constraints from right to left and the rows from bottom to top.

$$\begin{array}{c} \\ \text{row 4} \\ \text{row 3} \\ \text{row 2} \\ \text{row 1} \end{array} \begin{array}{c} \begin{array}{ccccc} C_5 & C_4 & C_3 & C_2 & C_1 \end{array} \\ \left[\begin{array}{ccccc} \text{W} & \text{L} & & & \\ & \text{W} & \text{L} & & \\ & & \text{W} & \text{L} & \\ & & & \text{W} & \text{L} \end{array}\right] \end{array} \quad \Longrightarrow \quad \begin{array}{c} \\ \text{row 4} \\ \text{row 3} \\ \text{row 2} \\ \text{row 1} \end{array} \begin{array}{c} \begin{array}{ccccc} \theta_5 & \theta_4 & \theta_3 & \theta_2 & \theta_1 \end{array} \\ \left[\begin{array}{ccccc} +1 & -1 & & & \\ & +1 & -1 & & \\ & & +1 & -1 & \\ & & & +1 & -1 \end{array}\right] \end{array} \tag{109}$$

If we fix, say, the weight θ_1 corresponding to constraint C_1, then there exists a unique weight vector that satisfies the condition $\langle \boldsymbol{\theta}, \overline{\mathbf{a}} \rangle = 1$ for every row $\overline{\mathbf{a}}$ of the derived diagonal HG-comparative tableau, namely the weight vector $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_n)$ constructed as in (110): we use the value of $\hat{\theta}_1$ and row 1 in order to conclude that $\hat{\theta}_2$ must exceed $\hat{\theta}_1$ by 1; then, we use this value of $\hat{\theta}_2$ and row 2 in order to conclude that $\hat{\theta}_3$ must exceed $\hat{\theta}_1$ by 2; and so on.

$$\begin{array}{lcccl} & & & & \hat{\theta}_1 \\ \text{row 1} & \Rightarrow & \hat{\theta}_2 - \hat{\theta}_1 = 1 & \Rightarrow & \hat{\theta}_2 = \hat{\theta}_1 + 1 \\ \text{row 2} & \Rightarrow & \hat{\theta}_3 - \hat{\theta}_2 = 1 & \Rightarrow & \hat{\theta}_3 = \hat{\theta}_2 + 1 = \hat{\theta}_1 + 2 \\ \text{row 3} & \Rightarrow & \hat{\theta}_4 - \hat{\theta}_3 = 1 & \Rightarrow & \hat{\theta}_4 = \hat{\theta}_3 + 1 = \hat{\theta}_1 + 3 \\ \text{row 4} & \Rightarrow & \hat{\theta}_5 - \hat{\theta}_4 = 1 & \Rightarrow & \hat{\theta}_5 = \hat{\theta}_4 + 1 = \hat{\theta}_1 + 4 \end{array} \tag{110}$$

In the general case, there is a unique weight vector $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_n)$ such that $\hat{\theta}_1$ is equal to a fixed value and furthermore $\langle \hat{\boldsymbol{\theta}}, \overline{\mathbf{a}} \rangle = 1$ for every row $\overline{\mathbf{a}}$ of the derived diagonal HG-comparative tableau, namely the weight vector (111).

[10] Let me explain why all rows are support vectors. Let $\overline{\mathbf{A}} \setminus \overline{\mathbf{a}}$ be the HG-comparative tableau obtained from $\overline{\mathbf{A}}$ by suppressing one of its row $\overline{\mathbf{a}}$. It is well known that a row $\overline{\mathbf{a}}$ is *not* a support vector iff it can be dropped without affecting HG-compatibility, namely the two tableaux $\overline{\mathbf{A}}$ and $\overline{\mathbf{A}} \setminus \overline{\mathbf{a}}$ are HG-compatible with exactly the same weight vectors. There is no single row of the HG-comparative tableau derived from a diagonal OT-comparative tableau that can be dropped without affecting HG-compatibility.

(111) $\hat{\theta}_k = \hat{\theta}_1 + (k-1) \qquad k = 1, 2, \ldots, n$

Thus, in order to solve the optimization problem (108) it is sufficient to solve the optimization problem (112) in the scalar decision variable θ_1, as the weight vector that solves (108) can then be reconstructed thorough (111) from the value of θ_1 that solves (112).

(112) $$\begin{array}{ll} \text{minimize:} & \displaystyle\sum_{k=1}^{n} \Big(\theta_1 + k - 1\Big)^2 \\ \text{subject to:} & \theta_1 \in \mathbb{R} \end{array}$$

As the objective function of the optimization problem (112) is (strictly) convex, the unique solution can be determined by setting its derivative to zero, which yields the solution (113).

(113) $\hat{\theta_1} = -\frac{1}{2}(n-1)$

In conclusion, the squared inverse of the margin μ can be computed as in (114). Here, I have reasoned as follows: in step (a), I have used the fact that $1/\mu^2$ coincides with the squared norm of the unique solution $\hat{\boldsymbol{\theta}}$ of the optimization problem (107), or equivalently (108); in step (b), I have used the fact that the latter vector can be described as in (111); in step (c), I have used the fact that the value $\hat{\theta}_1$ is provided by (113); the remaining identities are simple algebraic manipulations.

(114) $$\begin{aligned} \frac{1}{\mu^2} &= \|\hat{\boldsymbol{\theta}}\|^2 \\ &= \sum_{k=1}^{n} \Big(\hat{\theta}_1 + k - 1\Big)^2 \\ &= \sum_{k=1}^{n} \Big(-\frac{1}{2}(n-1) + k - 1\Big)^2 \\ &= \sum_{k=1}^{n} \Big(\frac{1}{4}(n-1)^2 + (k-1)^2 - (n-1)(k-1)\Big) \\ &= \frac{1}{4}n(n-1)^2 + \sum_{k=1}^{n-1} k^2 - (n-1)\sum_{k=1}^{n-1} k \\ &= \frac{1}{4}n(n-1)^2 + \frac{1}{6}n(n-1)(2n-1) - \frac{1}{2}n(n-1)^2 \\ &= \frac{1}{12}n(n^2-1) \end{aligned}$$

The claim thus follows from the upper bound R^2/μ^2 provided in (106), together with the identities $R^2 = 2$ and $1/\mu^2 = n(n^2-1)/12$. □

Claim 10 *The worst-case number of updates of the OT online algorithm (56) with the new cautious promotion-demotion update rule (79) run on Pater's comparative tableau of order n starting from the null initial vector grows with n at most at the order of n^5.* ■

Proof. Let me denote by $\overline{\mathbf{A}}_n$ the HG-comparative tableau derived through the mapping (104) from Pater's OT-comparative tableau $\mathbf{A}_n$ of order n. To illustrate, I provide in (115) the derived HG-comparative tableaux corresponding to the three Pater's tableaux in (75).

(115) $$\overline{\mathbf{A}}_4 = \begin{bmatrix} 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & 1 & -1 & \end{bmatrix} \quad \overline{\mathbf{A}}_5 = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & 1 & -2 & 1 & \\ & & & 1 & -1 & \end{bmatrix} \quad \overline{\mathbf{A}}_6 = \begin{bmatrix} 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & 1 & -2 & 1 & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -1 & \end{bmatrix}$$

Again as in the preceding proof, the worst case number of updates $T_{\text{OT}}(n)$ required by the OT online algorithm with the new cautious promotion/demotion update rule (66) on Pater's comparative tableau

$\mathbf{A}_n$ can be bound by R^2/μ^2, as in (106). The maximum norm R of the rows of $\overline{\mathbf{A}}_n$ is a constant, namely it does not depend on n. Again as in the preceding proof, the squared inverse $1/\mu^2$ of the margin coincides with the solution of the optimization problem (108), repeated in (116).

(116) minimize: $\|\boldsymbol{\theta}\|^2$
subject to: $\langle \boldsymbol{\theta}, \overline{\mathbf{a}} \rangle = 1$ for every row $\overline{\mathbf{a}}$ of Pater's derived tableau $\overline{\mathbf{A}}_n$

Suppose that the weight θ_1 is fixed to zero. Then the condition that $\langle \boldsymbol{\theta}, \overline{\mathbf{a}} \rangle = 1$ for every row $\overline{\mathbf{a}}$ of the derived Pater HG-comparative tableau univocally determines the weight vector $\boldsymbol{\theta}$. To illustrate, consider Pater's comparative tableau of order $n=5$, repeated in (117) together with the corresponding derived Pater HG-comparative tableau. For convenience, I have numbered the constraints from right to left and the rows from bottom to top.

(117)
$$\begin{array}{c} \\ \text{row 4} \\ \text{row 3} \\ \text{row 2} \\ \text{row 1} \end{array} \begin{array}{c} \begin{array}{ccccc} C_5 & C_4 & C_3 & C_2 & C_1 \end{array} \\ \left[\begin{array}{ccccc} \text{W} & \text{L} & \text{W} & & \\ & \text{W} & \text{L} & \text{W} & \\ & & \text{W} & \text{L} & \text{W} \\ & & & \text{W} & \text{L} \end{array}\right] \end{array} \Longrightarrow \begin{array}{c} \\ \text{row 4} \\ \text{row 3} \\ \text{row 2} \\ \text{row 1} \end{array} \begin{array}{c} \begin{array}{ccccc} \theta_5 & \theta_4 & \theta_3 & \theta_2 & \theta_1 \end{array} \\ \left[\begin{array}{ccccc} +1 & -2 & +1 & & \\ & +1 & -2 & +1 & \\ & & +1 & -2 & +1 \\ & & & +1 & -1 \end{array}\right] \end{array}$$

There exists a unique weight vector $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_5)$ such that $\hat{\theta}_1 = 0$ and furthermore $\langle \hat{\boldsymbol{\theta}}, \overline{\mathbf{a}} \rangle = 1$ for every row $\overline{\mathbf{a}}$ of the derived Pater HG-comparative tableau, namely the vector constructed as in (118): to start, we set $\hat{\theta}_1 = 0$; then, we use this value of $\hat{\theta}_1$ and row 1 in order to conclude that $\hat{\theta}_2$ must be equal to 1; then, we use these values of $\hat{\theta}_1$ and $\hat{\theta}_2$ and row 2 in order to conclude that $\hat{\theta}_3$ must be equal to 3; and so on.

(118)
$$\begin{array}{lllll} & & & & \hat{\theta}_1 = 0 \\ \text{row 1} & \Rightarrow & \hat{\theta}_2 - \hat{\theta}_1 = 1 & \Rightarrow & \hat{\theta}_2 = 1 \\ \text{row 2} & \Rightarrow & \hat{\theta}_3 - 2\hat{\theta}_2 + \hat{\theta}_1 = 1 & \Rightarrow & \hat{\theta}_3 = 3 \\ \text{row 3} & \Rightarrow & \hat{\theta}_4 - 2\hat{\theta}_3 + \hat{\theta}_2 = 1 & \Rightarrow & \hat{\theta}_4 = 6 \\ \text{row 4} & \Rightarrow & \hat{\theta}_5 - 2\hat{\theta}_4 + \hat{\theta}_3 = 1 & \Rightarrow & \hat{\theta}_5 = 10 \end{array}$$

In the general case, there is a unique weight vector $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_n)$ such that $\hat{\theta}_1 = 0$ and furthermore $\langle \hat{\boldsymbol{\theta}}, \overline{\mathbf{a}} \rangle = 1$ for every row $\overline{\mathbf{a}}$ of the derived Pater HG-comparative tableau, namely the weight vector defined by the recursion in (119a). It is trivial to prove by induction on k that the recursion (119a) can be made explicit as in (119b).

(119) a.
$$\begin{array}{rcll} \hat{\theta}_1 & = & 0 & \\ \hat{\theta}_2 & = & 1 & \\ \hat{\theta}_k & = & 1 + 2\hat{\theta}_{k-1} - \hat{\theta}_{k-2}, & \text{for } k = 3, \ldots, n \end{array}$$

b. $\hat{\theta}_k = \dfrac{k^2 - k}{2}$ for $k = 3, \ldots, n$

The squared inverse of the margin μ can thus be bounded as in (120), since $1/\mu^2$ coincides with the solution of the optimization problem (116), which is upper-bounded by the squared norm of the weight vector defined in (119b). The leading term of the expression obtained in (120) is $\sum_{k=1}^{n} k^4$ and thus the claim follows from the well-known identity $\sum_{k=1}^{n} k^4 = \frac{1}{5}n^5 + \frac{1}{2}n^4 + \frac{1}{3}n^3 - \frac{1}{30}n$.

(120) $$\frac{1}{\mu^2} \leq \|\hat{\boldsymbol{\theta}}\|^2 = \sum_{k=1}^{n} \left(\frac{k^2 - k}{2} \right)^2$$

It is easy to verify that the replacement of the solution of the optimization problem (116) with the squared norm of the feasible weight vector (119) does not substantially worsen the bound. □

A.5 Proof of claim 11

In this Subsection, I present a proof of claim 11, repeated below. The proof is actually just a slight variant of the proof of claim 2 presented above in Appendix A.2.

Claim 11 *Given an OT-comparative row and a weight vector $\boldsymbol{\theta}$, consider the HG-comparative row $\overline{\mathbf{a}} = (\overline{a}_1, \ldots, \overline{a}_n)$ derived from them according to (81), repeated below. Here, w is the total number of winner-preferrers (i.e. constraints that have a* W *in the row* $\mathbf{a}$*) and ℓ is the total number of loser-preferrers (i.e. constraints that have an* L *in the row* $\mathbf{a}$*) and are furthermore undominated w.r.t. $\boldsymbol{\theta}$ (i.e. there is no winner-preferrer with a larger weight).*

$$(121)\quad \overline{a}_k = \begin{cases} \ell & \text{if } a_k = \text{W} \quad \text{if } C_k \text{ is winner-preferrer} \\ -w & \text{if } a_k = \text{L} \quad \text{if } C_k \text{ is an undominated loser-preferrer} \\ 0 & \text{if } a_k = \text{E} \quad \text{otherwise} \end{cases}$$

If the OT-comparative row $\mathbf{a}$ is not OT-compatible with some ranking derived from the weight vector $\boldsymbol{\theta}$, then the corresponding derived HG-comparative row $\overline{\mathbf{a}}$ is not HG-compatible with $\boldsymbol{\theta}$ either. ■

Proof. Let W be the set of winner-preferrers, as in (122a); let L be the set of undominated loser-preferrers, as in (122b). The hypothesis that the OT-comparative row $\mathbf{a}$ is not OT-compatible with some ranking derived from the weight vector $\boldsymbol{\theta}$ ensures that L is not empty.

$$(122)\quad \text{a. } W = \left\{ k \,\middle|\, a_k = \text{W} \right\} \qquad \text{b. } L = \left\{ k \,\middle|\, a_k = \text{L},\ \theta_k \geq \max_{h \in W} \theta_h \right\}$$

The chain of inequalities (123) shows that the derived HG-comparative row $\overline{\mathbf{a}}$ is not HG-compatible with the weight vector $\boldsymbol{\theta}$. This chain of inequalities is analogous to the chain of inequalities in (97) used in Appendix A.2 to prove claim 2.

$$\begin{aligned} (123)\quad \sum_{h=1}^{n} \theta_h \overline{a}_h &\overset{(a)}{=} \sum_{h \in W} \theta_h \overline{a}_h + \sum_{h \in L} \theta_h \overline{a}_h + \sum_{h \notin W \cup L} \theta_h \overline{a}_h \\ &\overset{(b)}{=} \ell \sum_{h \in W} \theta_h - w \sum_{h \in L} \theta_h \\ &\overset{(c)}{\leq} \ell w \max_{h \in W} \theta_h - w \sum_{h \in L} \theta_h \\ &\overset{(d)}{\leq} \ell w \max_{h \in W} \theta_h - w \ell \min_{h \in L} \theta_h \\ &= \ell w \underbrace{\left(\max_{h \in W} \theta_h - \min_{h \in L} \theta_h \right)}_{(*)} \\ &\overset{(e)}{\leq} 0 \end{aligned}$$

Here, I have reasoned as follows: in step (a), I have split the set $\{1, \ldots, n\}$ that h runs over into the the set of winner-preferrers W in (122a), the set of undominated loser-preferrers L in (122b) and their complement; in step (b), I have used the definition (121) of the components $\overline{a}_1, \ldots, \overline{a}_n$ of the derived HG-comparative row; in step (c), I have used once more (95) to upper bound the sum $\sum_{h \in W} \theta_h$ with its biggest term $\max_{h \in W} \theta_h$ multiplied by the number w of terms; in step (d), I have reasoned analogously to lower bound the sum $\sum_{h \in L} \theta_h$ with its smallest term $\min_{h \in L} \theta_h$ multiplied by the number ℓ of terms; in step (e), I have used the hypothesis that the weight vector $\boldsymbol{\theta}$ admits a refinement which is not OT-compatible with the comparative row $\mathbf{a}$, which means in turn that there is a loser-preferrer whose weight is at least as large as the largest weight among winner-preferrers, so that the quantity (*) is non-positive. □

References

Bernhardt, Barbara Handford, and Joseph Paul Stemberger. 1998. *"Handbook of phonological development from the perspective of constraint-based nonlinear phonology"*. Academic Press.

Bertsimas, Dimitris, and John N. Tsitsiklis. 1997. *Linear Optimization*. Athena Scietific.

Boersma, Paul. 1997. "How We Learn Variation, Optionality and Probability". In *IFA Proceedings 21*, 43–58. University of Amsterdam: Institute for Phonetic Sciences.

Boersma, Paul. 1998. Functional Phonology. Doctoral Dissertation, University of Amsterdam. The Hague: Holland Academic Graphics.

Boersma, Paul. 2008. "Some Correct Error-driven Versions of the Constraint Demotion Algorithm". *Linguistic Inquiry* 40:667–686.

Boersma, Paul, and Bruce Hayes. 2001. "Empirical Tests for the Gradual Learning Algorithm". *Linguistic Inquiry* 32:45–86.

Boersma, Paul, and Clara Levelt. 2000. "Gradual Constraint-Ranking Learning Algorithm Predicts Acquisition Order". In *Proceedings of the 30th Child Language Research Forum*, 229–237. Standford University: CSLI. Corrected version (ROA 361, 1999/08/28).

Boersma, Paul, and Joe Pater. 2007. "Convergence Properties of a Gradual Learner for Harmonic Grammar". In *Proceedings of NELS 38*, –. .

Boersma, Paul, and Joe Pater. 2008. "Convergence Properties of a Gradual Learning Algorithm for Harmonic Grammar". Ms.

Cesa-Bianchi, Nicolò, and Gábor Lugosi. 2006. *Prediction, Learning, and Games*. Cambridge University Press.

Coetzee, Andries W., and Joe Pater. 2008. Weighted constraints and gradient restrictions on place co-occurrence in muna and arabic. *Natural Language and Linguistic Theory* 26:289–337.

Cristianini, Nello, and John Shawe-Taylor. 2000. *An Introduction to Support Vector Machines and Other Kernel-Based Methods*. Cambridge University Press.

Curtin, Suzanne, and Kie Zuraw. 2002. "Explaining Constraint Demotion in a Developing System". In *Proceedings of the 26th annual Boston University Conference on Language Development*, ed. Anna H.-J. Do, Laura Domínguez, and Aimee Johansen, –. Cascadilla Press. .

Dombi, József, Csanád Imreh, and Nándor Vincze. 2007. "Learning Lexicographic Orders". *European Journal of Operational Research* 183.2:748–756.

Dresher, E. 1999. "Charting the Learning Path: Cues to Parameter Setting". *Linguistic Inquiry* 30:27–67.

Fishburn, P. C. 1974. "Lexicographic Orders, Utilities and Decision Rules: A Survey". *Management Science* 20:1442–1471.

Gnanadesikan, Amalia E. 2004. "Markedness and Faithfulness Constraints in Child Phonology". In *Constraints in phonological acquisition*, ed. René Kager, Joe Pater, and Wim Zonneveld, 73–108. Cambridge: Cambridge University Press. Circulated since 1995.

Hayes, Bruce. 2004. "Phonological Acquisition in Optimality Theory: The Early Stages". In *Constraints in Phonological Acquisition*, ed. R. Kager, J. Pater, and W. Zonneveld, 158–203. Cambridge University Press.

Hayes, Bruce, and Colin Wilson. 2008. "A maximum entropy model of phonotactics and phonotactic learning". *Linguistic Inquiry* 39:379–440.

Jesney, Karen, and Anne-Michelle Tessier. 2007. "Re-evaluating learning biases in Harmonic Grammar". In *University of massachusetts occasional papers 36: Papers in theoretical and computational phonology*, ed. Michael Becker.

Jesney, Karen, and Anne-Michelle Tessier. 2008. "Gradual learning and faithfulness: consequences of ranked vs. weighted constraints". In *Proceedings of NELS38*, –.

Jesney, Karen, and Anne-Michelle Tessier. 2009. "Biases in Harmonic Grammar: the road to restrictive learning". *Natural Language and Linguistic Theory* .

Keller, Frank. 2000. Gradience in Grammar. Experimental and Computational Aspects of Degrees of Grammaticality. Doctoral Dissertation, University of Edinburgh.

Keller, Frank. 2005. "Linear Optimality Theory as a Model of Gradience in Grammar". In *Gradience in Grammar: Generative Perspectives*, ed. Gisbert Fanselow, Caroline Féry, Ralph Vogel, and Matthias Schlesewsky, –. Oxford: Oxford University Press.

Keller, Frank, and Ash Asudeh. 2002. "Probabilistic Learning Algorithms and Optimality Theory". *Linguistic Inquiry* 33.2:225–244.

Legendre, Géraldine, Yoshiro Miyata, and Paul Smolensky. 1990a. "Harmonic Grammar: A formal multi-level connectionist theory of linguistic well-formedness: An application". In *Proceedings of the twelfth annual conference of the Cognitive Science Society*, 884–891. Cambridge, MA: Lawrence Erlbaum.

Legendre, Géraldine, Yoshiro Miyata, and Paul Smolensky. 1990b. "Harmonic Grammar: A formal multi-level connectionist theory of linguistic well-formedness: Theoretical foundations". In *Proceedings of the twelfth annual conference of the Cognitive Science Society*, 388–395. Cambridge, MA: Lawrence Erlbaum.

Levelt, Clara C., Niels O. Schiller, and Willem J. Levelt. 2000. "The Acquisition of Syllable Types". *Language Acquisition* 8(3):237–264.

Magri, Giorgio. 2007. "The multiplicative GLA". Talk delivered at NECPhon 1, University of Massachusetts at Ahmerst.

Magri, Giorgio. 2010a. "A note on the equivalence between Recursive Constraint Demotion and the Fourier-Motzkin Elimination Algorithm". ENS manuscript.

Magri, Giorgio. 2010b. "The OT online model of the early stage of the acquisition of phonotactics: a computational perspective". Manuscript.

Pater, Joe. 2008. "Gradual Learning and Convergence". *Linguistic Inquiry* 39.2:334–345.

Pater, Joe. 2009. "Weighted Constraints in Generative Linguistics". *Cognitive Science* 33:999–1035.

Potts, Christopher, Joe Pater, Karen Jesney, Rajesh Bhatt, and Michael Becker. 2010. "Harmonic Grammar with Linear Programming: From linear systems to linguistic typology". *Phonology* 27(1):1–41.

Prince, Alan. 2002. "Entailed Ranking Arguments". ROA 500.

Prince, Alan, and Paul Smolensky. 2004. *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell. As Technical Report CU-CS-696-93, Department of Computer Science, University of Colorado at Boulder, and Technical Report TR-2, Rutgers Center for Cognitive Science, Rutgers University, New Brunswick, NJ, April 1993. Rutgers Optimality Archive 537 version, 2002.

Riggle, Jason. 2009. "The Complexity of Ranking Hypotheses in Optimality Theory". *Computational Linguistics* 35(1):47–59.

Stemberger, Joseph Paul, and Barbara Handford Bernhardt. 1999. "The Emergence of Faithfulness". In *The Emergence of Language*, ed. B. MacWhinney, 417–446. Mahweh, NJ: Erlbaum.

Stemberger, Joseph Paul, Barbara Handford Bernhardt, and Carolyn E. Johnson. 1999. "U-shaped learning in the acquisition of prosodic structure". Poster presented at the sixth International Child Language Congress.

Tesar, Bruce. 1995. "Computational Optimality Theory". Doctoral Dissertation, University of Colorado, Boulder. ROA 90.

Tesar, Bruce, and Paul Smolensky. 1998. "Learnability in Optimality Theory". *Linguistic Inquiry* 29:229–268.

Vapnik, Vladimir N. 1998. *Statistical Learning Theory*. John Wiley and sons.