The Learnability of the Accentuation of Sino-Japanese Words and Loanwords: The Hidden Structure Problem*

Motong Li
Osaka University

ABSTRACT. This paper analyzes the learnability of dominant accentual patterns of Sino-Japanese words and loanwords using the constraint ranking proposed by Li (2017) as the target grammar. Convergence results of computational simulations vary across algorithms and levels of inputs. Learning in the face of hidden structure, learners of Robust Interpretive Parsing (RIP; Tesar and Smolensky 2000) did not converge, indicating the failure to make full use of available probabilistic information. Instead, two novel parsing strategies proposed by Jarosz (2013) help to solve related problems of RIP, yielding significant improvements in performance even with the relatively complex target grammar.

Keywords: learnability, hidden structure, Gradual Learning Algorithm, Robust Interpretive Parsing, Expected Interpretive Parsing

1. Introduction

Learnability deals with the problem of how learners approach, and eventually converge to the target grammar. In Optimality Theory (OT; Prince and Smolensky 1993/2004) particularly, learnability asks how learners equipped with an initial constraint ranking update each constraint position until the final ranking matches the adult one. Compared to general OT work that focuses on the construction of the adult grammar, learnability is less studied, and less accountable when there are hidden structures like feet that need to be parsed (e.g. whether [σσ'σ] should be parsed into iambic /(σσ')σ/ or trochaic /σ(σ'σ)/). Depending on the data and the algorithm, the attempt to disambiguate sometimes fails, bringing inconsistency to the whole OT analysis.

Taking several combinations of learning and parsing algorithms into consideration, this paper conducts a computational study on the learnability of dominant accentual patterns observed in Sino-Japanese words and loanwords. The main purpose here is to examine the convergence rate of different algorithms with or without the intervention of the hidden structure problem, and to present various learning patterns into which the results can be classified.

2. Multi-level representation

Apoussidou (2007), which studied the learnability of metrical phonology in detail, used a three-level phonological representation for the hidden structure problem. As shown in (1), these three levels, Underlying Form (UF), Surface Form (SF) and Overt Form (OF), are represented by vertical bars, slashes and brackets respectively.

(1) Three-level phonological representation (Apoussidou 2007)

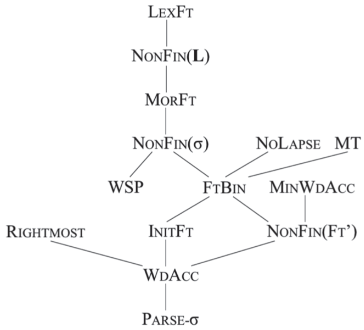
UF	|σσσ|	Syllables only
SF	/(σ'σ)σ/	Syllables, foot structures and the stress
OF	[σ'σσ]	Syllables and the stress

UF only consists of syllables, and can be extracted trivially from SF and OF in Apoussidou's analyses of grammatical stress in Latin and Pintupi. SF, which is the output level of general OT analyses, contains the full structural description, namely syllables, foot structures and the stress. Compared to SF, OF lacks the hidden foot structures but has the remaining overt information, and is hence more realistic considering the process of perception. Ambiguity emerges when OF is parsed into SF for learners to obtain the foot structure.

3. Target grammar and dominant accentual patterns

Based on Ito and Mester (2016) which conducted a thorough OT analysis of loanwords, Li (2017) proposed a constraint ranking that accounts for dominant accentual patterns of both Sino-Japanese words and loanwords simultaneously, as presented in (2).

(2) Constraint hierarchy as the target grammar[1] (Li 2017)



Dominant accentual patterns of the two lexical strata are shown in (3), with foot structure optimally specified (L: light syllable; H: heavy syllable; #: morpheme boundary; +: lexeme boundary; boldface: foothead).

(3) Dominant accentual patterns (Li 2017)

a. Sino-Japanese words and loanwords

(**L**’)	(**L**’L)	(**H**’)		

b. Sino-Japanese words

(**L**’)#L	(**L**)#(**L**L)	(**L**’L)#L	(**H**’)#L	(**L**)#(**H**)
(**H**)#(**L**L)	(**L**L)#(**H**)	(**H**)#(**H**)	(**L**L)#(**L**L)	

c. Loanwords

(**L**’L)L	(**H**’)L	(**L**’)H	(**L**L)(**L**L)	(**H**)(**L**L)
L(**H**’)L	(**L**L)(**L**’L)L	(**L**L)L(**L**’L)L	(**H**’)H	(**H**)L(**L**’L)L
(**H**)(**L**’)H	L(**L**’L)H	L(**H**’)H	L(**H**)(**L**L)	(**L**’L)H
(**H**)+(**H**)	(**L**L)+(**L**)	(**L**L)+(**H**)	(**H**)(**H**’)L	(**L**L)(**L**L) (**L**’L)L

In order to explain the different accentual distribution in these lexical strata, the analysis in Li (2017) highly relied on constraints such as quantity-sensitive NonFin(**L**) and morpho-phonological LexFt and MorFt, which all referred to the foot structure in particular ways. The complexity of foot specification makes it even harder to parse the hidden structure as expected in an ambiguous condition.

4. Algorithms

This section briefly introduces two learning algorithms, Error-Driven Constraint Demotion (EDCD; Tesar and Smolensky 2000) and Gradual Learning Algorithm (GLA; Boersma 1997, 1999; Boersma and Hayes 2001), and the parsing algorithm Robust Interpretive Parsing (RIP; Tesar and Smolensky 2000), which are all widely used in learnability-related literature.

4.1 EDCD

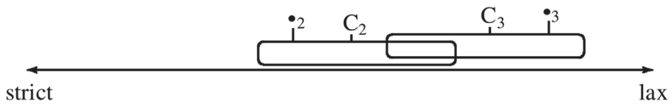
EDCD is a learning algorithm based on original OT which has discrete ranks. This algorithm assumes that, given a set of grammatical structural descriptions as inputs and a constraint hierarchy H_{start} as the initial grammar, learners use their current grammar H to select the optimal candidate, and compare it with the real data. If the current optimal output *loser* does not match the real data *winner*, then the deciding constraints that favor the *loser* should be minimally demoted below constraints that favor the *winner*, resulting in a new constraint hierarchy H_{new}. Learners will use H_{new} to update their current grammar and apply it to the next input. This learning process will not stop until all optimal outputs produced by the learner's grammar match the real data.

4.2 GLA

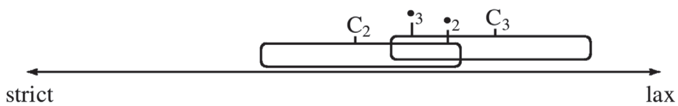
GLA is based on stochastic OT (Boersma 1997, 1999; Boersma and Hayes 2001), which is aimed at modeling free grammatical variation and speech errors that are frequently seen during the early learning stage. Unlike the discrete ranking in original OT, the ranking in stochastic OT is continuous, with each constraint numerically assigned a ranking value. At evaluation time, the evaluator adds a Gaussian random variable (mean = 0, standard deviation = *evaluation noise*) to each constraint to temporally randomize their current ranking value.

An example from Boersma and Hayes (2001) is shown in (4). Constraints C_2 and C_3 have their own distribution of ranking values. At evaluation time, it is possible to choose the selection points within the range of these two constraints. Generally as in (4a), selection point $\bullet_2$ is ranked higher than $\bullet_3$, resulting in $C_2 \gg C_3$ with high probability, but if $\bullet_2$ and $\bullet_3$ are randomized as in (4b), then C_3 would outrank C_2.

(4) Stochastic evaluation (Boersma and Hayes, 2001)
a. Common result: $C_2 \gg C_3$



b. Rare result: $C_3 \gg C_2$



Unlike EDCD, whose update rule only permits minimal demotion, GLA often demotes constraints that favor the *loser* and promotes constraints that favor the *winner* symmetrically. The range of demotion and promotion is decided by the parameter *plasticity* (ε). If ε is set to 1, then the ranking value should be added or subtracted by 1 regarding whether the constraint favor the *winner* or the *loser*. As reported in Boersma (2003), GLA performs better than EDCD when the problem of hidden structures gets involved[2].

4.3 RIP

RIP is a representative parsing algorithm in OT for learners to build hidden structures from overt information. An example from Apoussidou (2007) is shown in (5). Compared to general OT work whose input level is often UF, the input level of RIP is OF, which means that overt information like stress or accent is incorporated in it. Candidates in RIP are represented in SF, but all restricted to those whose OF is identical to the input. Learners then pick one candidate ☞$_{\text{RIP}}$ as optimal based on their current grammar (ALLFT-L ≫ ALLFT-R ≫ TROCHAIC ≫ IAMBIC in (5)), and interpret it as the *winner* for the current data.

(5) RIP in OT (Apoussidou 2007)

OF: [σσ’σ]	ALLFT-L	ALLFT-R	TROCHAIC	IAMBIC
a. /σ(σ’)σ/ [σσ’σ]	*!	*		
☞RIP b. /(σσ’) σ/ [σσ’σ]		*	*	
c. /σ(σ’σ)/ [σσ’σ]	*!			*

Next, as shown in (6), learners do general OT evaluation with UF as the input and an unlimited candidate set, pick one candidate ☞VP as optimal, and compare it with ☞RIP. If ☞VP and ☞RIP are different from each other, then the update rule based on EDCD (TROCHAIC demoted below IAMBIC) or GLA (TROCHAIC demoted and IAMBIC promoted numerically) is applied. If they match each other, then the current learning step ends and the algorithm moves to another datum.

(6) Error detection, given underlying form (Apoussidou 2007)

UF: |σσσ|	ALLFT-L	ALLFT-R	TROCHAIC	IAMBIC
a. |σσσ| /σ(σ’)σ/ [σσ’σ]	*!	*		
☞RIP b. |σσσ| /(σσ’)σ/ [σσ’σ]		*	*!	
c. |σσσ| /σ(σ’σ)/ [σσ’σ]	*!			*
☞VP d. |σσσ| /(σ’σ)σ/ [σ’σσ]		*		*
e. |σσσ| /σ(σσ’)/ [σσσ’]	*!		*	
f. |σσσ| /(σ’)σσ/ [σ’σσ]		**!		
g. |σσσ| /σσ(σ’)/ [σσσ’]	*!*			

In the next two sections, computational simulations are carried out to examine the convergence rate in different situations where the input level (SF or OF) and the learning algorithm (EDCD or GLA) are taken into consideration.

5. Simulations: without parsing

This section conducts simulations with SF as the input level, which means the *winners*’ optimal foot structures are known to learners without parsing. Training data are shown in (3) with full structural description. Praat (version 6.0.19; Boersma and Weenink 2016) is used to carry out all simulations in this paper.

5.1 EDCD

The parameter settings for EDCD learners are specified in (7). The update rule and the initial ε mean that ranking values of deciding constraints that favor the *loser* are minimally demoted below constraints that favor the *winner* by one. Due to the discreteness of original OT, the value of ε here does not affect the actual ranking. Because variation of ranking values is not allowed in EDCD, evaluation noise is set to 0. The number of ε assigned 1 indicates that ε is kept to 1 during the whole learning process. Replications per ε set to 1,000 stipulates that 1,000 learning steps are carried out for the current ε. After 1,000 learning steps, the result is counted as one simulation for one learner. Twenty learners are simulated here, and the initial ranking values of constraints are all set to 100. After the whole learning process, 100,000 SF data are randomly generated based on the learner’s final grammar, and are compared with the actual OF data shown in (3) to calculate the accuracy rate without considering the foot structure. The algorithm is considered to have converged successfully only if the learner’s accuracy rate is 100%.

(7) parameter settings for EDCD learners

Update rule	*Evaluation noise*	*Initial ε*	*Number of ε*	*Replications per ε*
EDCD	0	1	1	1,000

As a result, all EDCD learners converged to the target grammar (convergence rate = 100%), the effect of which was proven in Tesar and Smolensky (2000). Learners' final rankings can be classified into two patterns. Ten learners belong to the first pattern, Linguist Analysis (P_{LA}), which is identical to the target constraint hierarchy shown in (2). Figure 1 presents the ranking dynamics of this pattern. From this we can see that the learning process effectively stops at about the 125th learning step, after which ranking values do not change anymore.

The second pattern, Crucial Ties (P_{CT}; 10 learners), treats two or more constraints with tied ranks as one constraint and counts their violation marks jointly. An example for learners whose InitFt and WdAcc ranked equally is given in (8). The unaccented (8a) and ante-penultimate-mora-accented (8b) violate WdAcc and InitFt respectively. Because of the equal rank of these two constraints, violation marks here are canceled out, bringing the competition to the final-rank Parse-σ. However, as pointed out by Apoussidou (2007) and McCarthy (2008), the strategy of crucial ties does not have enough evidence to prove itself valid. Turning back to (8), it is implausible to consider that not parsing the initial syllable into a foot is as disadvantageous as not having an accent.

(8) Crucial Ties: WdAcc and InitFt

|LLLL| (/amerika/ 'America')	LexFt	MT	MinWdAcc	NoLapse	RightMost	NonFin(**L**)	MorFt	NonFin(σ)	FtBin	NonFin(Ft')	WSP	InitFt	WdAcc	Parse-σ
☞ a. /(**L**L)(**L**L)/													*	
b. /L(**L**'L)L/												*		*!*

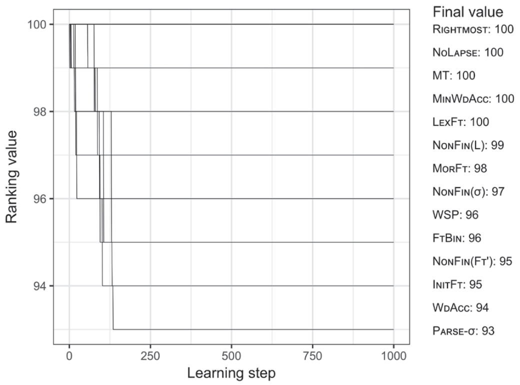


Figure 1 *EDCD, P_{LA}, without parsing*

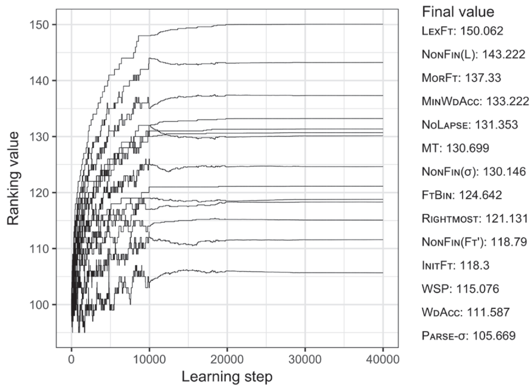


Figure 2 *GLA, P_{LA}, without parsing*

5.2 GLA

The parameter settings for GLA learners are specified in (9). The update rule set to *Symmetrical all* means that constraints that favor the *loser* are demoted, and constraints that favor the *winner* are promoted symmetrically. In order to make use of stochastic knowledge, evaluation noise is assigned a value of 2. The number of ε is set to 4 and each of them are used 10,000 times (40,000 times in total). To model different learning speeds in different learning stages, the decrement of ε is set to 0.1, which means that the first ε is 1, the second

0.1, the third 0.01, and the forth 0.001. Other parameter settings are the same as those in Section 5.1.

(9) parameter settings for EDCD learners

Update rule	*Evaluation noise*	*Initial ε*	*Number of ε*	*Replications per ε*	*ε decrement*
Symmetric all	2	1	4	10,000	0.1

GLA learners also learned the target grammar successfully (convergence rate = 100%), and they all belong to P_{LA}. The ranking dynamics of one representative learner are shown in Figure 2, where constraint demotion and promotion can both be observed. In the early learning stage, ε has a big value and learning errors occur frequently, which makes the ranking dynamics fluctuate greatly. But as learning proceeds, ε becomes smaller and learning errors decrease, resulting in near-stable ranking values.

6. Simulations: parsing with RIP

This section runs simulations with OF as the input level, which is more realistic considering the process of perception. In order to get a full structural description for the input, learners must make use of their parsing algorithm, at the risk of wrongly guessing a hidden structure that does not match the expected one. Here EDCD and GLA learners are equipped with RIP, with the same training data (foot structure removed) and parameter settings as in Section 5. Basic statistics of the accuracy rate of these two algorithms are given in (10).

(10) Descriptive statistics of EDCD/RIP and GLA/RIP

Algorithm	*N*	*Mean*	*SD*	*Median*	*Min*	*Max*
EDCD/RIP	20	0.89	0.15	0.98	0.5	0.98
GLA/RIP	20	0.92	0.01	0.92	0.92	0.94

Compared to the guaranteed convergence illustrated in Section 5, neither EDCD/RIP nor GLA/RIP learners converged to the target grammar. A statistical difference was not found between these two algorithms (Mann-Whitney U test, $U = 157.5$, $p = .245$).

6.1 EDCD/RIP

Results of EDCD/RIP learners can be classified into two patterns. The first pattern, $P_{NF(F')/IF\uparrow}$, has twelve learners with the reversed partial ranking NONFIN(FT'), INITFT ≫ FTBIN. The linear ranking of this pattern is shown in (11), with |LLLH| as its input. (11b) and (11c) have identical violation marks, hence the chance for each of them to be selected as optimal is nearly fifty-fifty. When (11c) is randomly chosen, the accuracy rate decreases because the OF of (11c) does not match the real data.

(11) $P_{NF(F')/IF\uparrow}$: antepenultimate- and pre-antepenultimate-mora accentuation

|LLLH| (/are'rugi:/ 'allergy')	INITFT	LEXFT	MT	MINWDACC	NONFIN(FT')	NOLAPSE	RIGHTMOST	NONFIN(**L**)	MORFT	NONFIN(σ)	WSP	FTBIN	WDACC	PARSE-σ
a. /L(**L**'L)H/	*!										*			**
☞ b. /(**L**)(**L**'L)H/											*	*		*
☞ c. /(**L**L)(**L**')H/											*	*		*

The remaining eight learners belong to P_{SLUMP}, whose ranking values of several constraints fall sharply. Accuracy rates of this pattern vary greatly, with a minimum of 50%. The ranking dynamics of one representative learner are shown in Figure 3, where the slump of FTBIN, PARSE-σ, RIGHTMOST and WDACC can be clearly observed. Due to the lowest rank of WDACC, words consisting of three or more moras will be output without an accent in this ranking, which cannot reflect the accented part in the real data correctly.

6.2 GLA/RIP

Most of the GLA/RIP learners have the reversed partial ranking MORFT ≫ NONFIN(**L**) and INITFT, NONFIN(FT') ≫ FTBIN ($P_{MF/NF(F')/IF\uparrow}$). This pattern also gives |LLLH| two optimal candidates, and wrongly outputs |H#L| (/kyuː'#yo/ 'salary') and |LL#L| (/ko'ku#chi/ 'notification') as /(**H**)#(**L**)/ and /(**L**L)#(**L**)/ without an accent. The ranking dynamics of this pattern are shown in Figure 4, where a fluctuating rise of WDACC and RIGHTMOST can be observed, with the ε decrement set to 0 (the result is the same as the ε decrement set to 0.1).

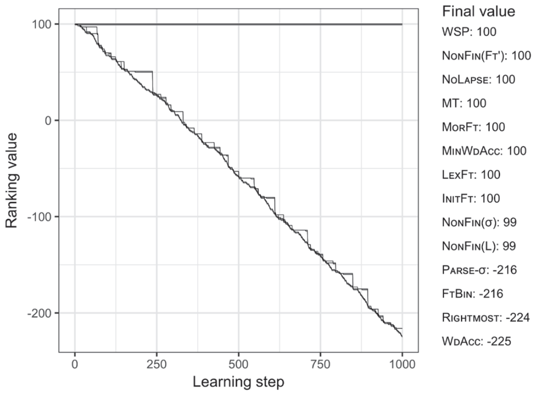


Figure 3 *EDCD/RIP, P_{SLUMP}*

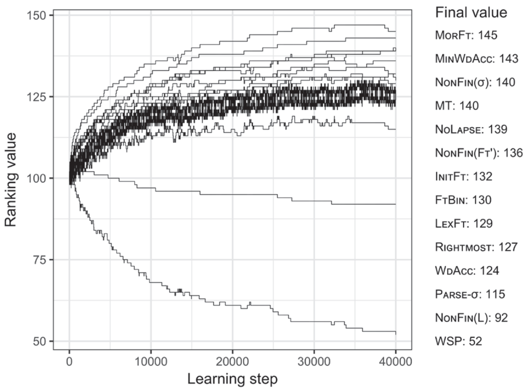


Figure 4 *GLA/RIP, $P_{MF/NF(F')/IF\uparrow}$ (ε decrement = 0)*

7. Reexamination of RIP

The convergence failure of EDCD/RIP and GLA/RIP learners can be partly attributable to the parsing algorithm. This section introduces Jarosz (2013), which gave a detailed re-examination of RIP and proposed two alternatives, Resampling Robust Interpretive Parsing (RRIP) and Expected Interpretive Parsing (EIP).

7.1 RRIP

In order to highlight the problem of RIP, the algorithm of RIP reformulated by Jarosz (2013) is shown in (12), which does not affect the behavior of the original parsing algorithm.

(12) Reformulated RIP for GLA (Jarosz 2013)
  1. Initialize Stochastic Grammar: G_0
  2. Iterate over *d* in *D*:
    a. Sample $G' \sim G_i$
    b. Input = uf(*d*)
    c. Output = Optimize$_{G'}$(Input)
    d. If overt(Output) ≠ *d*:
      i. Parse = RIP$_{G'}$(*d*)
      ii. G_{i+1} = Update(G_i, Parse, Output)

Jarosz stated that "[F]rom the reformulation it is clear that parsing is only relevant in case the selected grammar G′ generates an error…What is odd about this use of interpretive parsing in the stochastic setting, then, is that the learner nonetheless uses the known-to-be-incorrect G′ for interpretive parsing." In order to solve this problem, Jarosz added a simple modification to RIP, as shown in (13), with step 1 to 2c identical to those in (12). The difference between RRIP and RIP is that if G′ generates an error, the learner simply resamples another grammar G″ and uses it for interpretive parsing. This helps the learner reference its stochastic grammatical knowledge, the advantage of which is confirmed using simulations in Section 8.1.

(13) RRIP for GLA (Jarosz 2013)

…

2d. If overt(Output) ≠ *d*:

i. Sample G″ ~ G_i

ii. Parse = $RIP_{G''}(d)$

iii. G_{i+1} = Update(G_i, Parse, Output)

7.2 EIP

The second problem pointed out by Jarosz (2013), the parsing-production mismatch, is exemplified in (14). Given the assumed ranking values, ALLFT-L ≫ ALLFT-R ≫ TROCHAIC ≫ IAMBIC and ALLFT-R ≫ ALLFT-L ≫ TROCHAIC ≫ IAMBIC will be generated stochastically with equal probability, resulting in (14a) and (14c) being output as optimal nearly 50% of the time. TROCHAIC ≫ IAMBIC is fully activated to disfavor (14b) and (14d).

(14) The parsing-production mismatch (Jarosz 2013)

|LLL|	ALLFT-L 300	ALLFT-R 300	TROCHAIC 200	IAMBIC 100
a. /(**L**'L)L/		*		*
b. /(L**L**')L/		*	*	
c. /L(**L**'L)/	*			*
d. /L(L**L**')/	*		*	

However, if the learner hears [LL'L] and has to parse it with its current grammar, then candidates for RIP are restricted to (14b) and (14c) whose OF is identical to the real datum. This time, which candidate will be selected as the optimal SF depends in whole on the relative rank of ALLFT-R and ALLFT-L, leaving TROCHAIC and IAMBIC inactivated. In Jarosz's words, "[A]ccording to the learner's current grammar, /(L**L**')L/ is the only possible parse of [LL'L], but RIP fails to reflect this categorical restriction imposed by the grammar."

In order to solve this problem, Jarosz proposed EIP, as shown in (15). With step 1 to 2c unchanged, when an error is detected, EIP repeatedly resamples new grammars from the current one until the OF of the output matches the real datum *d.*

(15) EIP for GLA (Jarosz 2013)

…

2d. If overt(Output) ≠ *d*:

i. Parse ~ P(parse | G_i, *d*)

ii. G_{i+1} = Update(G_i, Parse, Output)

8. Simulations: parsing with RRIP and EIP

Based on Jarosz's proposition, this section conducts simulations using the novel parsing strategies RRIP and EIP. Because these algorithms require stochastic knowledge, the learning algorithm here is restricted to GLA. Training data and other parameter settings are the same as in Section 6. Basic statistics of the accuracy rate are given in (16), which includes the above-mentioned results of GLA/RIP for comparison.

(16) Descriptive statistics of GLA/RIP, GLA/RRIP and GLA/EIP

Algorithm	*N*	*Mean*	*SD*	*Median*	*Min*	*Max*
GLA/RIP	20	0.92	0.01	0.92	0.92	0.94
GLA/RRIP	20	0.94	0.02	0.94	0.94	1
GLA/EIP	20	1	0	1	1	1

A statistical difference is found between these three algorithms (Kruskal-Wallis H test, $H = 48.627$, $p < .001$), and the result for Bonferroni-corrected pairwise comparisons is significant (Mann-Whitney U test, $p < .001$ for all three pairs).

8.1 GLA/RRIP

Three patterns are observed in GLA/RRIP learners. Two learners belong to P_{LA}, converging to the target grammar successfully. The second pattern ($P_{\mathrm{MF/NF(F')}\uparrow}$; 17 learners) has the reversed partial ranking MORFT ≫ NONFIN(**L**) and NONFIN(FT') ≫ FTBIN, which wrongly parses |H#L| and |LL#L| into /(**H**)#(**L**)/ and /(LL)#(**L**)/ with WDACC and RIGHTMOST fluctuating relentlessly as presented in Figure 4. The last learner has the third pattern $P_{\mathrm{FB}\uparrow}$ where FTBIN outranks NONFIN(σ), NOLAPSE and MT, parsing |LH| (/ri'bon/ 'ribbon') and |HLH| (/rande'buː/ 'rendezvous') into /(**L**H)/ and /(**H**'L)H/ with a trimoraic foot.

To summarize, although two learners fortunately acquired the target grammar relying on their random resampling strategy, the fact that 90% of the learners failed to reach the goal still calls the validity of RRIP into question.

8.2 GLA/EIP

As shown in (16), all GLA/EIP learners successfully converged with their outputs all matching the real data in the OF level. Twelve learners are classified into P_{LA}, and the remaining eight learners belong to the pattern $P_{\mathrm{NF(F')}\uparrow}$ with NONFIN(FT') outranking FTBIN. The ranking dynamics of $P_{\mathrm{NF(F')}\uparrow}$ are shown in Figure 5, with the ε decrement set to 0. The only difference between $P_{\mathrm{NF(F')}\uparrow}$ and P_{LA} is the output of |LL| (/me'mo/ 'note'). In P_{LA}, |LL| is parsed into /(**L**'L)/ with a bimoraic foot structure due to the effect of FTBIN, while in $P_{\mathrm{NF(F')}\uparrow}$ /(**L**')L/ is output as optimal, leaving the word-final mora unparsed to avoid violating NONFIN(FT'), as shown in (17). However, because these two outputs have the same OF [L'L], learners of these two patterns can communicate without any hindrance. Hence the result of $P_{\mathrm{NF(F')}\uparrow}$ is also counted as converged. Moreover, the final value of NONFIN(FT') and FTBIN in $P_{\mathrm{NF(F')}\uparrow}$ are almost identical, so it is highly probable for $P_{\mathrm{NF(F')}\uparrow}$ to shift to P_{LA} as learning proceeds.

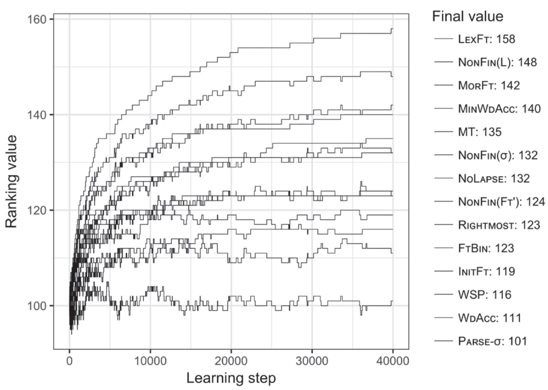


Figure 5 *GLA/EIP, $P_{\mathrm{NF(F')}\uparrow}$ (ε decrement = 0)*

(17) $P_{\mathrm{NF(F')}\uparrow}$: monomoraic-foot /(**L**')L/ as optimal

|LL| (/me'mo/ 'note')	LEXFT	NONFIN(**L**)	MORFT	MINWDACC	MT	NOLAPSE	NONFIN(σ)	NONFIN(FT')	RIGHTMOST	FTBIN	INITFT	WSP	WDACC	PARSE-σ
☞ a. /(**L**')L/										*				*
b. /(**L**'L)/								*!						

9. Conclusions

This study carried out several computational simulations to examine the learnability of dominant accentual patterns of Sino-Japanese words and loanwords. With a full structural description incorporated in the input level of perception, both EDCD and GLA learners converged efficiently. However, under more realistic conditions, the ambiguity of hidden structures cannot be effectively tackled by RIP. RRIP and EIP proposed by Jarosz (2013) were introduced to overcome the problem of RIP. Finally, GLA/EIP learners all successfully converged, showing that EIP can take full advantage of probabilistic information even in the face of an intricate target grammar. Future work will mainly cover the examination of different parameter settings and update rules, and the simultaneous learning-parsing algorithm for both the lexicon and the grammar.

Notes

[*] This paper is a revised version of a presentation at the Phonology Spring Meeting 2017, Keio University, on 23 June 2017. I really appreciate the participants who gave feedback to me, as well as two anonymous reviewers who commented on this paper.

[1] Definitions of constraints used in this paper are given below. LEXFT and NONFIN(**L**) were introduced in Li (2017), while other constraints were defined or referenced in Ito and Mester (2016).

LEXFT: Every lexeme minimally projects its own foot; MORFT: Every lexical morpheme minimally projects its own foot; MT: Feet are (**H**), (**L**L), and (**L**); NONFIN(σ): Word-final syllables are not footheads; NONFIN(**L**): Word-final light syllables are not footheads; NOLAPSE: Syllables are maximally parsed into feet; MINWDACC: A minimal prosodic word contains a prominence peak; RIGHTMOST: Violated by any foot following the head foot within the prosodic word; WSP: Heavy syllables are footheads; FTBIN: Feet are minimally binary at some level of analysis (mora, syllable); INTFT: A prosodic word begins with a foot; NONFIN(FT'): Violated by any head foot that is final in its PrWd; WDACC: A prosodic word contains a prominence peak; PARSE-σ: All syllables are parsed into feet.

[2] Tesar and Smolensky (2000) prepared 124 target grammars for their EDCD/RIP simulation, and only 75 of them were learned successfully (convergence rate = 60%). Using the same data set as Tesar and Smolensky (2000), Boersma (2003) reported that GLA/RIP converged on 70% of the target grammars.

References

Apoussidou, Diana. 2007. *The learnability of metrical phonology*. Ph.D. dissertation, University of Amsterdam.

Boersma, Paul. 1997. How we learn variation, optionality, and probability. *Proceedings of the Institute of Phonetic Sciences of the University of Amsterdam* 21.43–58.

Boersma, Paul. 1999. Optimality-Theoretic learning in the Praat program. *IFA proceedings* 23.17–35.

Boersma, Paul. 2003. Bruce Tesar and Paul Smolensky (2000). Learnability in Optimality Theory. Cambridge, Mass.: MIT Press. Pp. vii+ 140. *Phonology* 20.3.436–446.

Boersma, Paul and Bruce Hayes. 2001. Empirical Tests of the Gradual Learning Algorithm. *Linguistic inquiry* 32.1.45–86.

Boersma, Paul. and David Weenink. 2016. Praat: doing phonetics by computer [Computer program] Version 6.0.19. Online: http://www.praat.org/.

Ito, Junko and Armin Mester. 2016. Unaccentedness in Japanese. *Linguistic Inquiry* 47.3:471–526.

Jarosz, Gaja. 2013. Learning with hidden structure in Optimality Theory and Harmonic Grammar: Beyond Robust Interpretive Parsing. *Phonology* 30.1.27–71.

Li, Motong. 2017. The distribution of dominant accentual patterns in Sino-Japanese words: A comparison with loanwords. *Phonological Studies* 20.11–20.

McCarthy, John J. 2008. *Doing Optimality Theory: Applying theory to data*. Malden, MA: Wiley-Blackwell.

Prince, Alan and Paul Smolensky. 1993/2004. *Optimality Theory: Constraint interaction in generative grammar*. Malden, MA & Oxford, UK: Blackwell.

Tesar, Bruce and Paul Smolensky. 2000. *Learnability in Optimality Theory*. Cambridge, MA: The MIT Press.