# Performance of Learning & Learning from Performance

*Tamás Biró*

*ACLC*
*University of Amsterdam*

t.s.biro@uva.nl

Learning Meets Acquisition
Osnabrück, March 4, 2009

# Performance of Learning
# & Learning from Performance

Acquisition from performance:

- Child exposed to teacher's *performance*, not to *competence*.

Computationally: novel performance model producing learning data.

| $G_t$: competence of teacher | $\longrightarrow_{\mathrm{perf}}$ $L_t$ $\longrightarrow_{\mathrm{learn}}$ | $G_l$: competence of learner | $\longrightarrow_{\mathrm{perf}}$ $L_l$ |
|---|---|---|---|

Performance of learning:

- Influence of performance model: distance of $L_t$ and $L_l$?

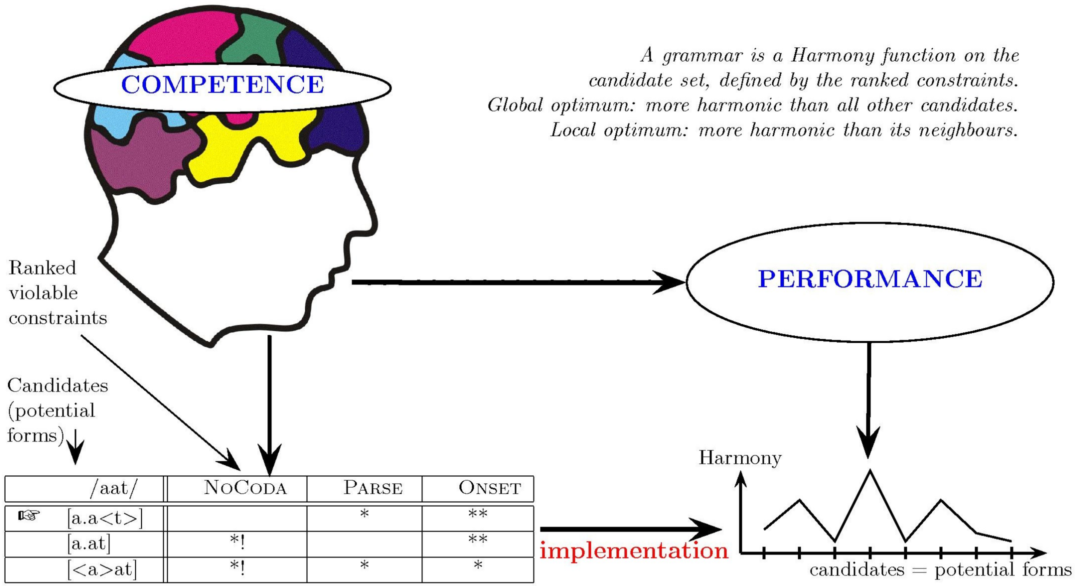
# Overview:

- Simulated Annealing for Optimality Theory: a performance model
- An example: string grammar
- Learning results with string grammar
- Measuring distance $L_t$ and $L_g$
- More learning results with string grammar

# Overview:

- Simulated Annealing for Optimality Theory: a performance model
- An example: string grammar
- Learning results with string grammar
- Measuring distance $L_t$ and $L_g$
- More learning results with string grammar

# *Competence* vs. *performance*

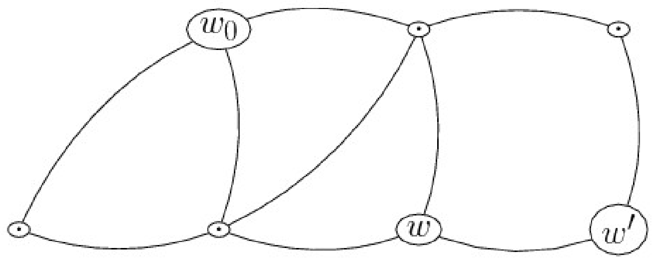


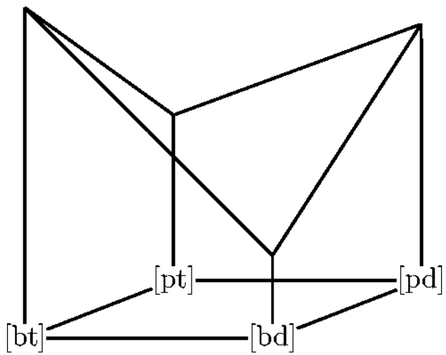

| /aat/ | NoCoda | Parse | Onset |
|---|---|---|---|
| ☞ [a.a<t>] | | * | ** |
| [a.at] | *! | | ** |
| [<a>at] | *! | * | * |

**Optimality Theory**
**grammar**
**competence model**
grammatical form = ☞ (globally) optimal candidate

**SA-OT**
**implementation**
**performance model**
produced forms = globally or locally optimal candidates

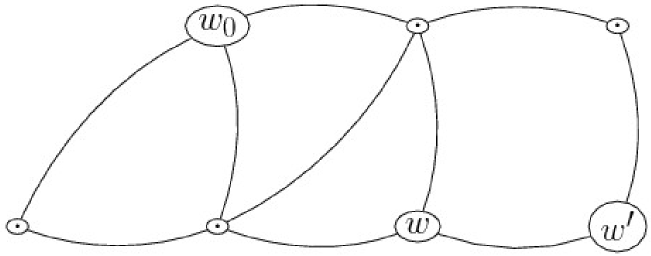
# Performance models (simulated annealing) for OT

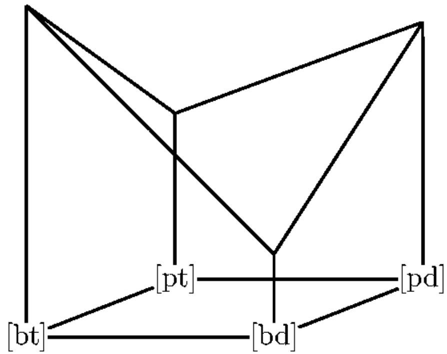



- Goal: to find the (globally) optimal candidate.
- Add a *neighbourhood structure* to the candidate set.
- Landscape's vertical dimension $=$ harmony.
- Neighbourhood structure $\rightarrow$ local optima.

# Performance models (simulated annealing) for OT




- Random walk. If neighbour more optimal: move. If less optimal: move early in the algorithm, don't move later.
- System can get stuck in local optima: errors produced.
- Precision of the algorithm depends on its speed (!!).

# Errors and irregularities

ICS (Smolensky & Legendre 2006), SA-OT (Bíró 2006): both implement Optimality Theory with *simulated annealing*.

- ☞ *Grammatical forms* = globally optimal
- ! *Performance errors*: frequency diminishes at slow (careful) production (as in traditional simulated annealing).
- ∼ *Irregularities*: frequency does not diminish at slow (careful) production (due to *strict domination*).

# Overview:

- Simulated Annealing for Optimality Theory: a performance model
- An example: string grammar
- Learning results with string grammar
- Measuring distance $L_t$ and $L_g$
- More learning results with string grammar

# Example: string-grammar 1

- *Candidates:* $\{0, 1, ..., P-1\}^L$
  E.g., $L = P = 4$: $0000$, $0001$, $0120$, $0123$,... $3333$.

- *Neighbourhood structure:* $w$ and $w'$ neighbours iff one basic step transforms $w$ to $w'$.

- Basic step: change exactly one character $\pm 1$, $\mod P$ (cyclicity). Neighbours of $0022$: $1022$, $0012$, $0322$,...

- Each neighbour with equal probability.

# Example: string-grammar 2

Markedness Constraints ($w = w_0 w_1 ... w_{L-1}$, $0 \leq n < P$):

- No-$n$: $*n(w) := \sum_{i=0}^{L-1}(w_i = n)$
- No-initial-$n$: $*\textsc{Initial}n(w) := (w_0 = n)$
- No-final-$n$: $*\textsc{Final}n(w) := (w_{L-1} = n)$
- Assimilation $\textsc{Assim}(w) := \sum_{i=0}^{L-2}(w_i \neq w_{i+1})$
- Dissimilation $\textsc{Dissim}(w) := \sum_{i=0}^{L-2}(w_i = w_{i+1})$

# Example: string-grammar 3

- Faithfulness to UR $\sigma$:

$$\textsc{Faith}_{\sigma}(w) = \sum_{i=0}^{L-1} d(\sigma_i, w_i)$$

where the distance of two characters:
$d(\sigma_i, w_i) = \min\left((a-b) \mod P, (b-a) \mod P\right)$.

# Example: string-grammar 4

$\mathcal{H}$: no0 $\gg$ ass $\gg$ Faith$_{\sigma=0000}$ $\gg$ ni1 $\gg$ ni0 $\gg$ ni2 $\gg$ ni3 $\gg$ nf0 $\gg$ nf1 $\gg$ nf2 $\gg$ nf3 $\gg$ no3 $\gg$ no2 $\gg$ no1 $\gg$ dis

Output frequencies for different $t_{step}$ (=inverse speed) values:

| *output* | t_step $= 1$ | t_step $= 0.1$ | t_step $= 0.01$ | t_step $= 0.001$ |
|---|---|---|---|---|
| ☞ 3333 | 0.1174 ± 0.0016 | 0.2074 ± 0.0108 | 0.2715 ± 0.0077 | 0.3107 ± 0.0032 |
| ∼ 1111 | 0.1163 ± 0.0021 | 0.2184 ± 0.0067 | 0.2821 ± 0.0058 | 0.3068 ± 0.0058 |
| ∼ 2222 | 0.1153 ± 0.0024 | 0.2993 ± 0.0092 | 0.3787 ± 0.0045 | 0.3602 ± 0.0091 |
| ! 1133 | 0.0453 ± 0.0018 | 0.0485 ± 0.0038 | 0.0328 ± 0.0006 | 0.0105 ± 0.0014 |
| ! 3311 | 0.0436 ± 0.0035 | 0.0474 ± 0.0054 | 0.0344 ± 0.0021 | 0.0114 ± 0.0016 |
| ! others | 0.5608 | 0.1776 | $<$ 0.0002 | – |

$L = P = 4$, $T_{max} = 3$, $T_{min} = 0$, $K_{step} = 1$. $w_0$: each of the $4^4$ candidates 4 times.

Globally optimal form: ☞ 3333. But 13 local optima: 2222, $\{1,3\}^4$.

# Overview:

- Simulated Annealing for Optimality Theory: a performance model
- An example: string grammar
- Learning results with string grammar
- Measuring distance $L_t$ and $L_g$
- More learning results with string grammar

# Learning

Learning algorithms in Optimality Theory:

- Off-line learning algorithms: Recursive Constraint Demotion
  - Initial grammar from observations in pre-linguistic infants?
  - Produces typical "children errors": extra local optima
- On-line learning algorithms: Error Driven Constraint Demotion, Gradual Learning Algorithm.
  - Grammar improving gradually in childhood?

# Learning

Assumptions and heuristics behind learning algorithms:

- Traditional OT: observed form is optimal.
- SA-OT: observed form is locally optimal.
- Moreover: more frequent form is more harmonic.
  Not always true in trad. OT; even less true in SA-OT.

Nevertheless, some success!

# Same performance, different competence after RCD

| target | after RCD | after GLA |
|---|---|---|
| No0 15 | No0 15 | No0 15.000000 |
| Ass 14 | Ass 12 | Ass 14.000004 |
| Fai 13 | Fai 4 | Fai 6.100000 |
| Ni1 12 | Ni1 8 | Ni1 10.400004 |
| Ni0 11 | Ni0 13 | Ni0 13.000000 |
| Ni2 10 | Ni2 5 | Ni2 7.100000 |
| Ni3 9 | Ni3 3 | Ni3 -1.500000 |
| Nf0 8 | Nf0 14 | Nf0 14.000000 |
| Nf1 7 | Nf1 10 | Nf1 6.300000 |
| Nf2 6 | Nf2 6 | Nf2 8.100000 |
| Nf3 5 | Nf3 2 | Nf3 3.600000 |
| No3 4 | No3 7 | No3 3.000000 |
| No2 3 | No2 11 | No2 13.100004 |
| No1 2 | No1 9 | No1 10.900006 |
| Dis 1 | Dis 1 | Dis -1.000000 |

Constraint name + its rank.

| | target | after RCD | after GLA |
|---|---|---|---|
| | 315 2222 | 322 2222 | 298 2222 |
| | 210 1111 | 221 1111 | 238 1111 |
| ☞ | 196 3333 | 200 3333 | 225 3333 |
| | 55 3111 | 53 1133 | 54 3311 |
| | 49 1133 | 50 3111 | 45 1133 |
| | 48 1333 | 45 1113 | 41 1333 |
| | 48 3331 | 45 3311 | 40 1113 |
| | 46 1113 | 42 3331 | 37 3331 |
| | 42 3311 | 32 1333 | 35 3111 |
| | 4 1331 | 4 1131 | 3 1331 |
| | 4 3133 | 4 1331 | 3 3113 |
| | 3 3113 | 2 1311 | 2 3133 |
| | 2 1311 | 2 3113 | 2 3313 |
| | 2 3313 | 2 3133 | 1 1131 |

Absolute frequency + output form.

# GLA corrects children speech errors

| target | after RCD | after GLA |
|---|---|---|
| No0 15 | No0 2 | No0 12.500014 |
| Ass 14 | Ass 14 | Ass 12.299995 |
| Fai 13 | Fai 7 | Fai -0.500000 |
| Ni1 12 | Ni1 9 | Ni1 10.800001 |
| Ni0 11 | Ni0 15 | Ni0 15.000000 |
| Ni2 10 | Ni2 13 | Ni2 11.499998 |
| Ni3 9 | Ni3 11 | Ni3 10.699999 |
| Nf0 8 | Nf0 8 | Nf0 13.300017 |
| Nf1 7 | Nf1 6 | Nf1 7.900000 |
| Nf2 6 | Nf2 1 | Nf2 -12.200008 |
| Nf3 5 | Nf3 3 | Nf3 9.000000 |
| No3 4 | No3 4 | No3 3.700000 |
| No2 3 | No2 12 | No2 11.400000 |
| No1 2 | No1 5 | No1 3.300000 |
| Dis 1 | Dis 10 | Dis 11.700005 |

| target | after RCD | after GLA |
|---:|---:|---:|
| 302 2222 | 279 1111 | 292 2222 |
| 235 1111 | 277 2222 | 230 1111 |
| 208 3333 | 277 3333 | 194 3333 |
| 49 1113 | 39 1133 | 73 3311 |
| 49 3311 | 39 3311 | 52 1133 |
| 45 1333 | 38 2200 | 47 1113 |
| 44 1133 | 37 3111 | 45 3111 |
| 40 3111 | 35 1333 | 38 1333 |
| 37 3331 | 2 1113 | 37 3331 |
| 5 1331 | 1 1000 | 5 1331 |
| 4 3313 | | 4 3133 |
| 2 3113 | | 3 3313 |
| 2 3133 | | 2 1311 |
| 1 1131 | | 2 3113 |
| 1 1311 | | |

GLA decreases freq. of 1111 and 3333. “Child form” 2200: extra local optimum.

# GLA does not converge towards target

| target | after RCD | after GLA |
|---|---|---|
| No0 15 | No0 11 | No0 20.200031 |
| Ass 14 | Ass 15 | Ass 21.200026 |
| Fai 13 | Fai 8 | Fai 7.600000 |
| Ni1 12 | Ni1 7 | Ni1 5.299999 |
| Ni0 11 | Ni0 5 | Ni0 13.300017 |
| Ni2 10 | Ni2 4 | Ni2 7.500000 |
| Ni3 9 | Ni3 10 | Ni3 -0.100000 |
| Nf0 8 | Nf0 14 | Nf0 19.300018 |
| Nf1 7 | Nf1 12 | Nf1 0.599994 |
| Nf2 6 | Nf2 3 | Nf2 9.500005 |
| Nf3 5 | Nf3 2 | Nf3 1.600000 |
| No3 4 | No3 1 | No3 -14.000012 |
| No2 3 | No2 13 | No2 18.900017 |
| No1 2 | No1 9 | No1 7.400000 |
| Dis 1 | Dis 6 | Dis -0.200000 |

| target | after RCD | after GLA |
|---|---|---|
| 300 2222 | 232 1111 | 250 1111 |
| 231 1111 | 216 2222 | 239 2222 |
| ☞ 214 3333 | 207 3333 | 226 3333 |
| 53 1133 | 201 0000 | 170 0000 |
| 51 3311 | 49 1133 | 36 2200 |
| 50 3331 | 34 3311 | 33 1133 |
| 46 1333 | 30 0022 | 31 0022 |
| 38 3111 | 29 2200 | 30 3311 |
| 33 1113 | 14 1113 | 3 1333 |
| 6 1331 | 11 3331 | 2 1113 |
| 1 1131 | 1 1333 | 2 3111 |
| 1 3113 | | 2 3331 |

Constraint ranks diverge. 0000 is locally optimal.

# Overview:

- Simulated Annealing for Optimality Theory: a performance model
- An example: string grammar
- Learning results with string grammar
- Measuring distance $L_t$ and $L_g$
- More learning results with string grammar

# During online learning

How close the learner is to the target/teacher?

- Language: UF mapped to set of (form, frequency $P(f)$) pairs ($\sum_f P(f) = 1$).
- Target language: distribution $P_t(f)$.
- Learner's language: distribution $P_l(f)$.

How to measure distance of $P_l$ from $P_t$?

# During online learning

Relative entropy (a.k.a. Kullback-Leibler divergence, etc.):

$$D(P\|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

Jensen-Shannon divergence:

$$JSD(P\|Q) = \frac{D(P\|M) + D(Q\|M)}{2}$$

where $M(x) = \frac{P(x)+Q(x)}{2}$.

# During online learning

Properties of the Jensen-Shannon divergence:

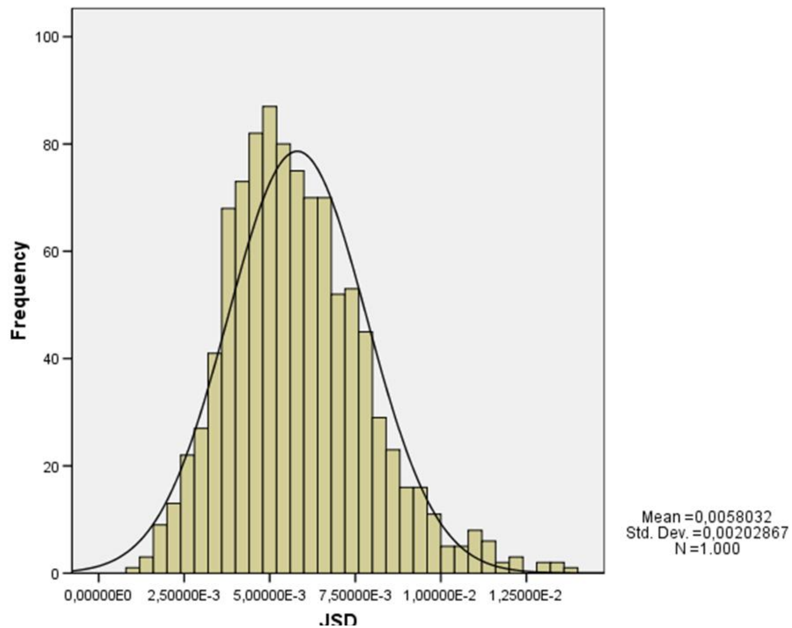
- Symmetric: $JSD(P\|Q) = JSD(Q\|P)$.
- Non-negative: $JSD(P\|Q) \geq 0$. $JSD(P\|Q) \leq 1$.
- $JSD(P\|Q) = 0$ if and only if $P(x) = Q(x), \ \forall x$.
  $JSD(P\|Q) = 1$ if and only if $P(x) \cdot Q(x) = 0, \ \forall x$.

Same language: $JSD(L_t\|L_l) = 0$.
Not a single overlap: $JSD(L_t\|L_l) = 1$.

# During online learning

Same grammar producing two finite samples: how different can they be?

JSD of two samples of $N = 1000$ each, produced by the target string grammar. Repeat $1000$ times. $JSD < 0.015$ $(JSD = 0.006 \pm 0.002)$.



# Overview:

- Simulated Annealing for Optimality Theory: a performance model
- An example: string grammar
- Learning results with string grammar
- Measuring distance $L_t$ and $L_g$
- More learning results with string grammar

# During online learning

Number of learning steps needed to reach $JSD < 0.015$:

61 71 81 91

101 111 121 141 151 161 161 171 171 181 181 191 191

201 211 211 271 271 281 281 281 281 291

301 321 331 341 351 351 361 381 391

401 411 411 471 511 551 591 661 661 691

711 741 791 831 871 971 991

1001 1001 1001 1011 1231 1241 1291

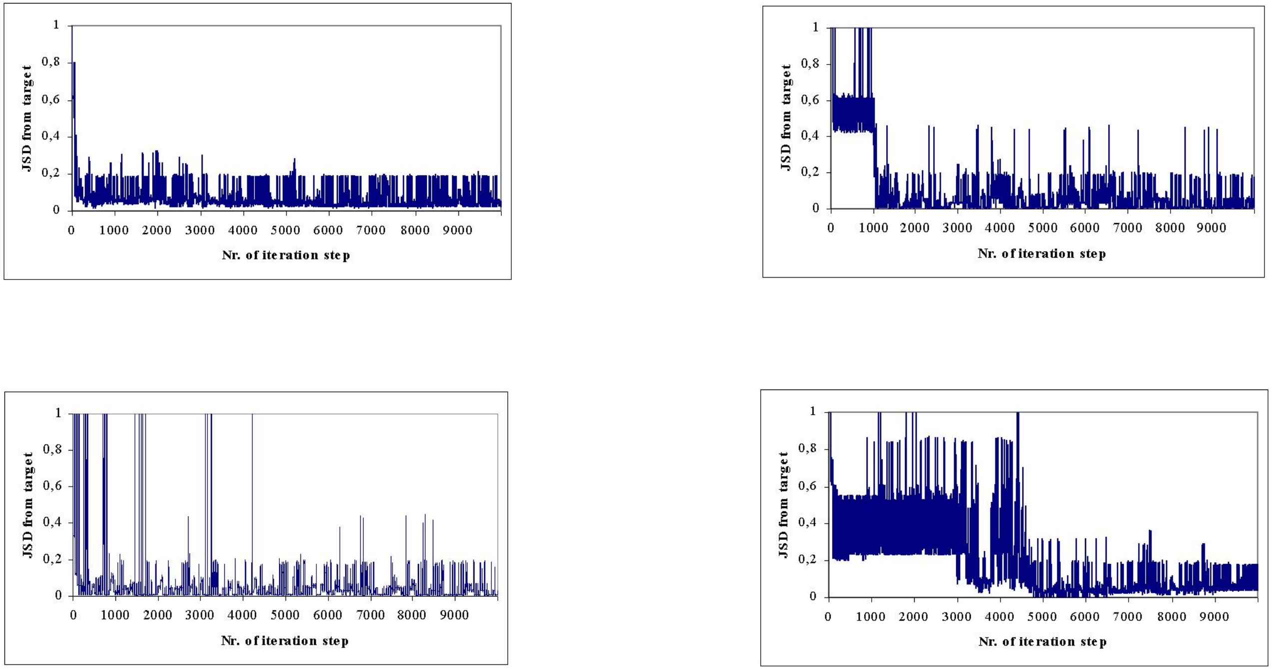
1411 1431 1631 1671 1761 1781 1831 1961

2001 2061 2081 2301 2321 2641 3221 3291 3491 3651 3891

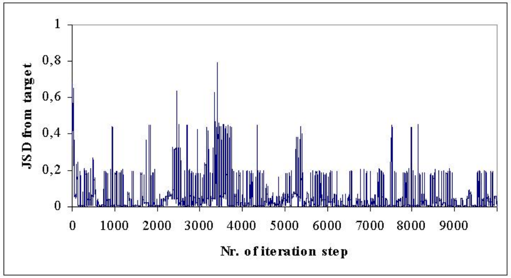
4491 4811 5031 5051 5241 5251 5491 5551 6461 6931 8171 8411

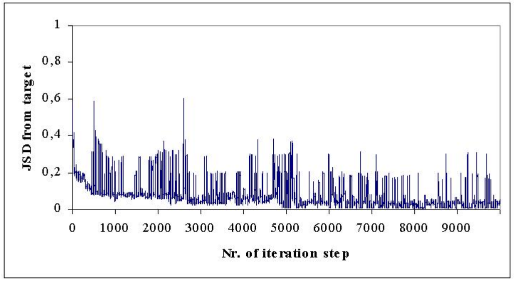
10651 12301 14771 17281 18041 24031 78841 103411 174871

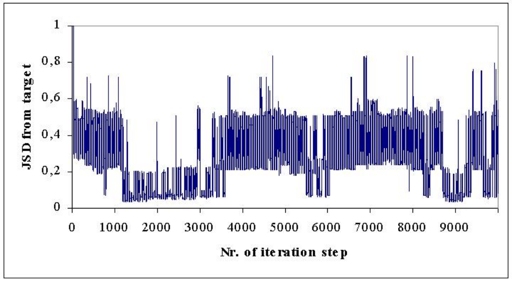
# Learning curves

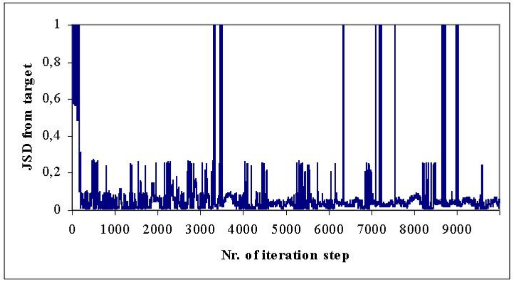


# Learning curves



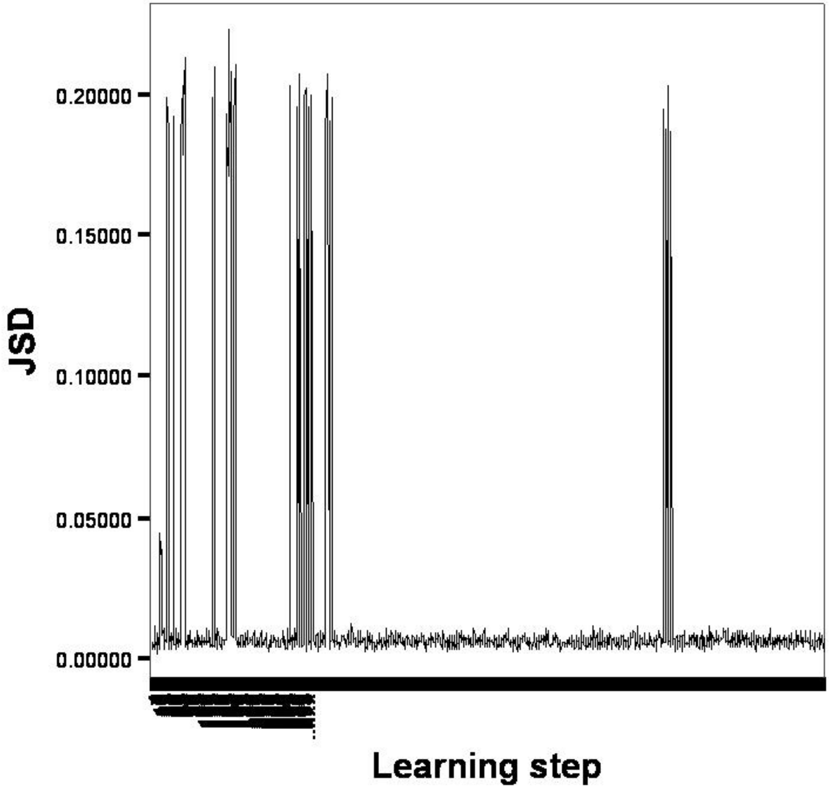



# During online learning

After having reached the target?

Learning if initially learner's grammar same as teacher's grammar:



# Conclusions

- Simulated Annealing for OT:
  a model of linguistic performance.
- Learning algorithms for SA-OT: much left to be done.
- Jensen-Shannon divergence:
  measuring how close $L_l$ is from $L_t$.
- Evolution: long tail and outbursts cause some children acquire a slightly different language.

Thank you for your attention!

Tamás Biró

birot@nytud.hu

# The SA-OT algorithm

```
w := w_init ;
for K = K_max to K_min step K_step
      for t = t_max to t_min step t_step
             CHOOSE   random w' in neighbourhood(w) ;
             COMPARE  w' to w:  C := fatal constraint
                                d := C(w') - C(w);
             if d <= 0 then w := w';
             else           w := w' with probability
                       P(C,d;K,t) = 1         , if C < K
                                  = exp(-d/t) , if C = K
                                  = 0         , if C > K
      end-for
end-for
return w
```