Supplementary Materials for *Beyond plain and extra-grammatical morphology: echo-pairs in Hungarian*

Márton Sóskuthy[a], Péter Rácz[b]

[a] *University of British Columbia*
[b] *Cognitive Development Center, Central European University*

1. Introduction

This document provides an overview of our MGL model fitting process, the parameterisation of the GCM adopted in the main paper and the procedure for estimating parameter values for the GCM. The main paper provides more detail on extracting predictions from the MGL and the GCM. The code for defining the models, estimating the parameters and generating prediction probabilities is available online at the following address:

`https://github.com/petyaracz/SoskuthyRacz2018`

2. The Minimal Generalisation Learner

The main paper provides a description of the key features of the MGL, including how it constructs and weights rules and how it uses these weighted rules in predicting outputs. Here we focus on (i) the parameter settings we used with the MGL and (ii) how we transformed the inputs in order to make the most of the MGL algorithm.

2.1. MGL parameter settings

We used the default parameter settings of the MGL implementation available here:

`http://www.mit.edu/~albright/mgl/`

The lower and upper confidence limit parameters of the MGL were both set to 0.75. The lower confidence limit parameter determines the extent to which the generality of a rule should affect its confidence score. With low values around 0.5, accuracy is much more heavily weighted than generality (so e.g. a rule that predicts correct outputs for 4 out of 5 forms would receive a confidence score that is similar to a rule that predicts correct outputs for 400 out of 500 forms); with higher values, generality becomes increasingly important, so a rule covering only 5 forms will be weighted down compared to a rule that covers 500. The upper confidence limit parameter is used to protect against the MGL proposing certain rules that are overly general (see Albright and Hayes 2002 for a more detailed description).

We also fit a version of the MGL with a lower confidence limit of 0.55 and an upper confidence limit of 0.75, and another version with the same values set to 0.75 and 0.95 respectively, but these did not improve on the performance of the MGL with the default settings.

As noted in the main text, the MGL was able to propose feature-based rules using a modified version of the feature matrix in Albright and Hayes (2003). The feature 'impugnment' was also enabled – this allows the model to avoid making certain overgeneralisations, and relates directly to the upper confidence limit parameter mentioned above (Albright and Hayes, 2002).

2.2. MGL input transformation

We faced two issues when fitting the MGL to our data without altering the inputs.

1. The specification of the structural change of MGL rules does not use features, i.e. rules are always of the form $\emptyset \rightarrow \text{ɪd} \; / \; \ldots$, $\text{k} \rightarrow \text{m} \; / \; \ldots$, etc. but never of the form [+cons] → m / . . . . This leads to problems with our data: echo-pair formation typically consists in changing the initial segment of the base, and so there is at least one separate rule for each unique word-initial segment in our corpus; the MGL cannot abstract away general rules of the form C → m and C → b (where C is any consonant). This is especially problematic given that some of the nonce-forms have word-initial onsets that did not occur in the corpus data, and therefore the set of rules generated by the MGL does not contain any rules that apply to them. As a consequence, the MGL does not generate any predictions for such forms.

Email addresses: `marton.soskuthy@ubc.ca` (Márton Sóskuthy), `raczp@ceu.edu` (Péter Rácz)

2. While the MGL can include non-local segments in the structural description of its rules, it cannot make feature-based generalisations on the basis of such segments. Thus, the MGL cannot come up with a rule that has the structural description __ V_0 [+voi]: rules can only have one segment that is specified in terms of features on each side of the focus line, and any material intervening between the focus line and the feature-based specification must be in the form of specific segments (e.g. the vowel ɛ, but not V_0). This is problematic since some key generalisations about the data concern segments that are separated from the structural change by at least one segment (e.g. voiceless V_1C).

In order to overcome these issues, we transformed the inputs (i.e. the base forms) and the outputs (i.e. echos) as follows. We added an initial zero segment (marked as =) to bases that start with a vowel, and then swapped the first vowel of the base swapped with the second consonant. This means that the base forms *cica* and *izé* are represented as *ccia* and *=zié*. The echos were simply formed by taking the transformed base and adding the echo behaviour between the initial segment (a consonant or =) and the first vowel, yielding *cmcia* and *=bzié*.

The net effect of these transformations is that (i) echo-pair formation now involves the insertion of segments instead of morphing a segment into another (and therefore can be represented using general rules of the form $\emptyset \rightarrow \mathrm{m} \,/\, \ldots$), and (ii) both O_1 and O_2 are available for feature-based generalisations.

3. The Generalised Context Model

The Generalised Context Model (or GCM; Nosofsky 1986, 1988) is a model of categorisation that implements the notion of instance-based learning. The GCM assumes that categorisation is based on similarity to individual memories (or exemplars) rather than abstract generalisations. For instance, the GCM predicts that an English speaker faced with the task of creating a past tense form for a novel English verb (e.g. *spling*; Rácz et al. 2014) would base their decision not on a small set of explicit rules, but on a comparison with phonologically similar existing verbs (e.g. *sing* → *sang*; *ring* → *rang*). The GCM has been shown to outperform other models of categorisation in cases where the target categories are not linearly separable (Nosofsky 1986, 1988), and when the data contains small pockets of regularity which are not well captured by broad generalisations (Stemberger and MacWhinney 1988; Dawdy-Hesterberg and Pierrehumbert 2014; Rácz et al. 2014).

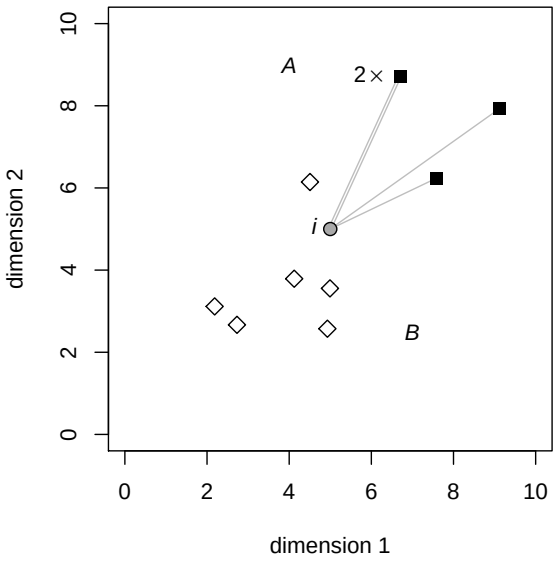

Figure 1: Comparison of a new item (*i*) to previously experienced items from two categories (*A* = filled squares vs. B = empty diamonds). The lines represent similarity comparisons to items in category *A*. One of the stored items occurs twice.

In the following paragraphs, we present a brief outline of the main components of the GCM, with particular focus on the roles of different parameters. In order to make the discussion easier to follow, the parameters will be referred to using the notation $\textsc{parameter}_p$, where the text in small capitals is the name of the parameter, and the text in the subscript is the name of the corresponding variable used in the formulae.

In this paper, the GCM is used for classification (as opposed to identification; cf. Nosofsky 1988) of items. The main task of the GCM is to calculate the conditional probabilities of different category labels given an item *i*, that is, determine how likely it is that *i* comes from categories *A*, *B*, etc. This is done by comparing *i* to stored representations of previously experienced items from different categories. Figure 1 illustrates this procedure. In this example, category *A* is represented by four items (one of which occurs twice), and category *B* by six items. The probability that *i* is from category *B* is higher than the probability that *i* is from category *A*, since *B* contains more items and these are more similar to *i* than the ones in category *B*. This intuition can be formalised as follows (this is a slightly modified version of the formulae provided in Ashby and Maddox 1993

and Nosofsky and Zaki 2002):

$$p(A \mid i) = \frac{\beta_A (\sum_{j \in A} N_j^\tau \eta_{ij})^\gamma}{\sum_{X \in C} [\beta_X (\sum_{k \in X} N_k^\tau \eta_{ik})^\gamma]} \quad (1)$$

Lowercase letters indicate specific items, uppercase letters indicate category labels and C indicates the set of all category labels. As before, i is the item that needs to be classified. Each category X has an associated prior probability, β_X, which we will refer to as the BASELINE$_\beta$. If the BASELINE$_\beta$ is high for category X relative to the other categories, X will have a high categorisation probability even if its overall similarity to i is low. N_j is the token frequency of item j. If an item occurs more than once in a category (as in Figure 1), each token contributes separately to the overall similarity (this is represented by the double line in Figure 1). The model presented here departs from the formulation in Ashby and Maddox (1993) by raising N_j to the power of τ. This parameter will be referred to as TOKEN WEIGHT$_\tau$, and it determines the importance of token frequency in the model. The value of TOKEN WEIGHT$_\tau$ is related to the role of token frequency in the following way:

- TOKEN WEIGHT$_\tau$ = 0: token frequency plays no role in categorisation;
- 0 < TOKEN WEIGHT$_\tau$ < 1: token frequency plays a diminished role in categorisation;
- TOKEN WEIGHT$_\tau$ = 1: the contribution of each type is proportional to its token frequency;
- TOKEN WEIGHT$_\tau$ > 1: token frequency plays an exaggerated role in categorisation.

The symbol η_{ij} stands for the similarity between items i and j. The higher the similarity between the two items, the more likely it is that i will be categorised as an instance of the category j belongs to. The details of the similarity metric used in this paper will be described in the next paragraph. Finally, the summed similarity of i to all members of category A is raised to the power of γ. γ controls the amount of determinism in decisions about category membership, and will be referred to as DETERMINISM$_\gamma$:

- DETERMINISM$_\gamma$ = 0: categorisation depends solely on the BASELINE$_\beta$ values, and is otherwise completely random;
- 0 < DETERMINISM$_\gamma$ < 1: categorisation depends on the overall similarities as well, but the outcomes are less predictable than we would expect based on the similarities;

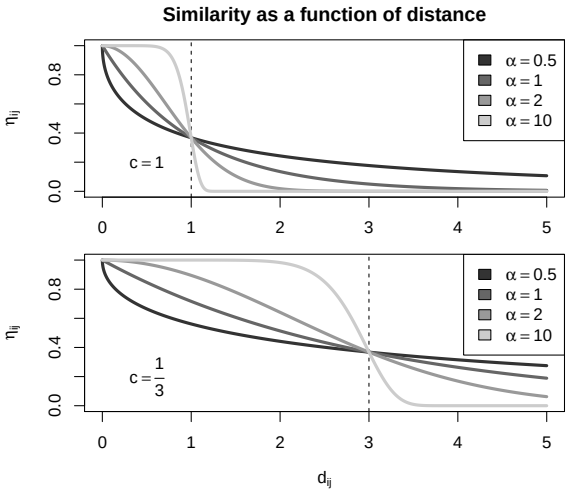

Figure 2: Similarity (η_{ij}) plotted as a function of distance (d_{ij}). The value of the SCALE$_c$ parameter is varied within the plots, whereas the value of ABRUPTNESS$_\alpha$ is varied between the plots.

- DETERMINISM$_\gamma$ = 1: if the BASELINE$_\beta$ values are the same, the conditional probability of category A is proportional to its overall similarity to i;
- DETERMINISM$_\gamma \to \infty$: assuming that the BASELINE$_\beta$ values remain unchanged, the categorisation decisions become deterministic; the category with the highest overall similarity to i will always be chosen as the outcome.

Note that the denominator in (1) is just a normalising constant which is the same for all categories. In other words, the conditional probabilities of the category labels are proportional to the numerator in (1).

The following formula describes the calculation of the similarity values, η_{ij}:

$$\eta_{ij} = e^{-(c\, d_{ij})^\alpha} \quad (2)$$

That is, the similarity between items i and j decreases as their distance (d_{ij}) increases. The distance metric used in this paper will be described in the next paragraph. The relationship between similarity and distance is modulated by two further parameters: c, a scaling constant (SCALE$_c$) and α, a parameter that controls the abruptness of the transition between distance values that yield high similarity and distance values that yield low similarity (ABRUPTNESS$_\alpha$). The effects of changing these two parameter settings are illustrated in Figure 2. Parameter SCALE$_c$ can be regarded as a measure of the sensitivity of the similarity metric: at high values of SCALE$_c$, even small changes in distance result in large changes

in similarity. The other parameter, ABRUPTNESS$_\alpha$, determines the extent to which the similarity metric behaves as a window function: as $\alpha \to \infty$, the similarity metric becomes a simple window function, which returns a value of 1 when $d_{ij} < 1/c$, and 0 when $d_{ij} > 1/c$ (the cut-off for the window function is indicated by the dotted line in Figure 2).

The distance between i and j, d_{ij}, is based on the differences between the individual features used to describe i and j (see the main paper for a description of the feature vectors we use to represent words that undergo echo-pair formation). However, not all features contribute equally to d_{ij}: some of them may contribute more information to the distance measure than others. For instance, using the segment-based representation introduced in the main paper, it is reasonable to assume that the second onset (O_2) is more important for the distance metric than the second nucleus (N_2), since O_2 is more relevant to echo-pair formation than N_2. The calculation of the distance metric can be formalised as follows (using the so-called *Minkowski r-metric*; cf. Nosofsky 1986):

$$d_{ij} = \Big[\sum_{k}^{F} w_k \, \mathrm{diff}(x_{ik}, x_{jk})^r \Big]^{1/r} \tag{3}$$

F represents the overall number of features, and w_k represents the feature weight for the kth feature (FEATURE WEIGHT$_{w_k}$). The symbols x_{ik} and x_{jk} represent the kth element of the feature vectors for i and j, respectively. The function $\mathrm{diff}(x_{ik}, x_{jk})$ returns a value of 0 when $x_{ik} = x_{jk}$ and a value of 1 when $x_{ik} \neq x_{jk}$. The parameter r determines the nature of the distance metric: when $r = 1$, d_{ij} is equivalent to the city-block (or Manhattan) metric, and when $r = 2$, d_{ij} is equivalent to the Euclidean metric (Nosofsky, 1986). This parameter will be referred to as METRIC TYPE$_r$, and is allowed to take on values between 1 and 2, essentially interpolating between the two different metric types.

We provide a summary of the parameters in Table 1 for convenience.

PARAMETER	DESCRIPTION
BASELINE$_{\beta_X}$	Baseline probability of category X
TOKEN WEIGHT$_\tau$	The importance of token frequency in calculating the similarity metric
DETERMINISM$_\gamma$	The extent to which the model makes deterministic predictions
SCALE$_c$	The precision of the similarity metric
ABRUPTNESS$_\alpha$	The extent to which the similarity metric behaves as a window function
FEATURE WEIGHT$_{w_k}$	The importance of the kth feature in calculating the distance metric
METRIC TYPE$_r$	A parameter that interpolates between different distance metrics (city-block vs. Euclidean)

Table 1: A summary of the parameters used in the current implementation of the GCM.

3.1. Estimation of GCM parameters

In this paper, the GCM is used to predict the probabilities of different echo behaviours for a variety of forms based on an existing corpus of echo-pairs. These predictions are made with the help of an `R` package called `rgcm` developed by the first author of this paper (Sóskuthy, 2015). `rgcm` relies heavily on another computationally implemented version of the GCM called `BayesGCM` (Vanpaemel, 2009).

As explained in the paper, two different types of GCM models are used: a naive and an informed version. The naive GCM uses a fixed set of values for the parameters described in the previous section, while the informed GCM estimates some of these parameters from the data. Similar to `BayesGCM`, parameter estimation is performed using JAGS, a platform-independent Gibbs sampler (Plummer, 2003).

Parameter estimation with JAGS follows a Bayesian logic. Each parameter to be estimated is given a prior distribution representing values that are deemed possible or likely based on previous knowledge (which could come from theory or from prior experience with the parameter in question). The sampling algorithm explores the posterior distribution for the parameters, which represents plausible values for the parameters taking into account both the data and the priors. Estimating the posterior distribution is much like estimating coefficients in a regression model, with three main differences: (1) the model is allowed to be much more complicated than a conventional regression model; (2) there is flexibility in specifying the priors for the model; and (3) the estimation process treats the parameters as random variables, returning a distribution of parameter values instead of a single-point estimate.

For the purposes of obtaining prediction probabilities, we simply use the expected value (i.e. the mean) of the posterior distribution. In other words, although the estimation process returns a distribution of values, these are subsequently reduced to a single number (as the goal here is not to make statistical inferences about the parameters of the model, but to provide prediction probabilities that can be subsequently included as predictors in statistical models).

The models in this paper use so-called uninforma-

PARAMETER	NAIVE GCM	INFORMED GCM
BASELINE$_{\beta_X}$	p(m) = p(b) = 0.5	ddirch($\alpha_{1,2}$ = 1)
TOKEN WEIGHT$_\tau$	0	0
DETERMINISM$_\gamma$	1	1
SCALE$_c$	13.06	13.06
ABRUPTNESS$_\alpha$	1	1
FEATURE WEIGHT$_{w_k}$	1/6 per feature	ddirch($\alpha_{1,2,...,6}$ = 1)
METRIC TYPE$_r$	1	1

Table 2: The parameter values and priors used in the naive and informed GCM.

tive (or flat) priors, which do not make any substantive hypotheses about the parameter values. Table 2 summarises the fixed parameter values used in the naive GCM and the prior distributions for the parameter values estimated in the informed GCM. The value of the parameter SCALE$_c$ is actually based on a previous model fit using `rgcm`, and is kept fixed here to allow more stable estimates for the other parameters. `ddirch` represents a so-called Dirichlet distribution, which is used when a given parameter is vector-valued (i.e. consists of multiple related sub-parameters) and the sum of the values in the vector is constrained to be 1. The Dirichlet distribution is defined by parameters α_1, α_2, ..., α_k, where k corresponds to the dimensionality of the vector (in this case, 2 for BASELINE$_{\beta_X}$ and 6 for FEATURE WEIGHT$_{w_k}$).

References

Albright, A., Hayes, B., 2002. Modeling English past tense intuitions with minimal generalization. In: SIGPHON 6: Proceedings of the Sixth Meeting of the ACL Special Interest Group in Phonology. pp. 58–69.

Albright, A., Hayes, B., 2003. Rules vs. analogy in English past tenses: A computational/experimental study. Cognition 90 (2), 119–161.

Ashby, F. G., Maddox, W. T., 1993. Relations between prototype, exemplar and decision bound models of categorization. Journal of Mathematical Psychology 37, 372–400.

Dawdy-Hesterberg, L. G., Pierrehumbert, J. B., 2014. Learnability and generalisation of arabic broken plural nouns. Language, Cognition and Neuroscience 29 (10), 1268–1282.

Nosofsky, R. M., 1986. Attention, similarity and the identification-categorization relationship. Journal of Experimental Psychology: General 115, 39–57.

Nosofsky, R. M., 1988. Similarity, frequency and category representations. Journal of Experimental Psychology: Learning Memory and Cognition 14, 54–65.

Nosofsky, R. M., Zaki, S. R., 2002. Exemplar and prototype models revisited: Response strategies, selective attention, and stimulus generalization. Journal of Experimental Psychology: Learning, Memory, and Cognition 28 (5), 924–940.

Plummer, M., 2003. JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. In: Proceedings of the 3rd international workshop on distributed statistical computing. Vol. 124. Vienna, Austria.

Rácz, P., Beckner, C., Hay, J. B., Pierrehumbert, J. B., 2014. 'rules', 'analogy' and social factors codetermine past-tense formation patterns in english. Workshop between SIGMORPHON and SIGFSM, Association for Computational Linguistics 2014, Baltimore, Maryland, June 22–27, 2014.

Sóskuthy, M., 2015. RGCM: Bayesian parameter estimation and prediction for the Generalised Context Model in R. Available from https://github.com/soskuthy/rgcm.

Stemberger, J. P., MacWhinney, B., 1988. Are inflected forms stored in the lexicon? In: Hammond, M., Noonan, M. (Eds.), Theoretical morphology: Approaches in modern linguistics. Academic Press, San Diego, CA, pp. 101–116.

Vanpaemel, W., 2009. BayesGCM: Software for Bayesian inference with the generalized context model. Behaviour Research Methods 41, 1111–1120.