

Syntactic Structures and Code Parameters

Kevin Shu · Matilde Marcolli

Received: 30 November 2016 / Revised: 15 February 2017 / Accepted: 22 February 2017 / Published online: 24 March 2017
© Springer International Publishing 2017

Abstract We assign binary and ternary error-correcting codes to the data of syntactic structures of world languages and we study the distribution of code points in the space of code parameters. We show that, while most codes populate the lower region approximating a superposition of Thomae functions, there is a substantial presence of codes above the Gilbert–Varshamov bound and even above the asymptotic bound and the Plotkin bound. We investigate the dynamics induced on the space of code parameters by spin glass models of language change, and show that, in the presence of entailment relations between syntactic parameters the dynamics can sometimes improve the code. For large sets of languages and syntactic data, one can gain information on the spin glass dynamics from the induced dynamics in the space of code parameters.

Keywords Natural languages · Syntactic parameters · Error-correcting codes · Code parameters · Asymptotic bounds · Spin glass dynamics

Mathematics Subject Classification 91F20 · 94B65 · 82C20

1 Introduction

This is a companion paper to [13], where techniques from coding theory were proposed as a way to address quantitatively the distribution of syntactic features across a set (or family) of languages. In this paper we perform a computational analysis, based on the syntactic structures of world languages recorded in the SSWL database [21], and on data of syntactic parameters collected by Longobardi and collaborators. We first analyze all codes obtained from pairs and triples of languages in the SSWL database and we show that their code points tend to populate the lower region of the space of code parameters, approximating two scaled copies of the Thomae function. Points that fall in that region behave essentially like random codes. We then analyze arbitrary subsets of languages and syntactic data from the SSWL database and we compute the density of the distribution of their code points, showing that,

K. Shu · M. Marcolli (✉)

Division of Physics, Mathematics, and Astronomy, California Institute of Technology, 1200 E. California Blvd, Pasadena, CA 91125, USA
e-mail: matilde@caltech.edu

K. Shu

e-mail: kshu@caltech.edu

while most of them populate the lower region, there is a significant presence of codes above the Gilbert–Varshamov (GV) bound and even above the asymptotic bound and the Plotkin bound. We consider spin glass models of language change and the induced dynamics on the space of code parameters, and show that, in the presence of entailment relations the dynamics can enter the region above the Gilbert–Varshamov bound. We also show how the induced dynamics in the space of code parameters can be helpful in gaining information on the behavior of the spin glass model on large datasets of languages and parameters, where convergence becomes very slow and a direct analysis of the dynamics becomes computationally difficult.

2 Codes and Code Parameters from Syntactic Structures

In [13] a new coding theory approach to measuring entropy and complexity of a set of natural languages was proposed. The idea is to associate to each language in the set a vector of binary variables that describe syntactic properties of the language. The notion of encoding syntactic structures through a set of binary syntactic parameters is a crucial part of the Principles and Parameters program of Linguistics developed by Chomsky, [3,4], see also [1] for an expository introduction. Thus, given a set of languages $\mathcal{L} = \{\ell_1, \dots, \ell_N\}$ and a set of n binary syntactic variables, whose values are known for all the N languages, one obtains a code $\mathcal{C}_{\mathcal{L}}$ in \mathbb{F}_2^n consisting of N code words.

As argued in [13], one can use the properties of the resulting codes $\mathcal{C}_{\mathcal{L}}$, as an error correcting code, and its position in the space of code parameters to measure how syntactic features are distributed across the languages in the set. Moreover, the position of the code $\mathcal{C}_{\mathcal{L}}$ in the space of code parameters, with respect to curves such as the Gilbert–Varshamov bound and the asymptotic bound provide a measure of entropy and of complexity of the set of languages \mathcal{L} , which differs from measures of entropy/complexity for an individual language.

2.1 Code Parameters and Bounds

In the theory of error-correcting codes (see for instance [18]), to a given code $\mathcal{C} \subset \mathbb{F}_q^n$, one assigns two *code parameters*: the *transmission rate*, or *relative rate* of the code, which measures how good the encoding is, and which is given by the ratio

$$R(\mathcal{C}) = \frac{k}{n}, \quad \text{with } k = \log_2(\#\mathcal{C}) = \log_2(N),$$

where k is the *absolute rate* of \mathcal{C} , and the *relative minimum distance* of the code, which measures how good the decoding is, and which is given by the ratio

$$\delta(\mathcal{C}) = \frac{d}{n}, \quad \text{with } d = \min_{\ell_1 \neq \ell_2 \in \mathcal{C}} d_H(\ell_1, \ell_2),$$

where $d_H(\ell_1, \ell_2)$ denotes the Hamming distance. For q -ary codes, the Hamming distance is the number of coordinates in which the two code words ℓ_1 and ℓ_2 differ. In the case of binary codes this is the same as

$$d_H(\ell_1, \ell_2) = \sum_{i=1}^n |x_i - y_i|,$$

for $\ell_1 = (x_i)_{i=1}^n$ and $\ell_2 = (y_i)_{i=1}^n$ in \mathcal{C} , with $x_i, y_i \in \{0, 1\}$. Codes \mathcal{C} that have both $R(\mathcal{C})$ and $\delta(\mathcal{C})$ as large as possible are optimal for error-correction, as they have more efficient encoding and less error-prone decoding: indeed, a larger δ means that there are fewer code words close to a received word, which makes it easier to identify the correct transmitted word. In general, it is not possible to arbitrarily improve both parameters, hence the quality of a code \mathcal{C} is estimated by the position of its code point $(\delta(\mathcal{C}), R(\mathcal{C}))$ in the space of code parameters of coordinates (δ, R) inside the square $[0, 1] \times [0, 1]$. Various bounds on code parameters have been studied [8, 18–20].

As discussed in [13] there are two bounds, that is, two curves in the space of code parameters, that have an especially interesting meaning: the Gilbert–Varshamov curve, which is related to the statistical behavior of random codes (see [2,5]), and the asymptotic bound, whose existence was proved in [8], further studied in [9–12]. The asymptotic

bound separates the region where code points are dense and have infinite multiplicity from the region where they are sparse and with finite multiplicity [9, 12]. The Gilbert–Varshamov curve for a q -ary code has a simple form,

$$R = 1 - H_q(\delta), \quad \text{with } H_q(\delta) = \delta \log_q(q-1) - \delta \log_q \delta - (1-\delta) \log_q(1-\delta),$$

the q -ary Shannon entropy. With the asymptotic bound, however, the situation is much more complicated, as one does not have an explicit expression. Indeed, the question of the computability of the asymptotic bound was posed in [9], and addressed in [12] in terms of a relation to Kolmogorov complexity (which is not a computable function). Namely, it is shown in [12] that the asymptotic bound becomes computable, given an oracle that can order codes by increasing Kolmogorov complexity. Even though one does not have an explicit expression for the asymptotic bound, several estimates on its location in the space of code parameters are described in [18]. Thus, in practical cases, it will be possible to obtain sufficient conditions to check if a code point violates the asymptotic bound by using some of these estimates. In particular, there is a relation between the asymptotic curve $R = \alpha_q(\delta)$ and the Gilbert–Varshamov bound,

$$\alpha_q(\delta) \geq 1 - H_q(\delta),$$

and a relation between the asymptotic bound and the Plotkin bound $R \leq 1 - \frac{\delta}{q}$. The Plotkin line lies above the Gilbert–Varshamov curve and the asymptotic bound satisfies $\alpha_q(\delta) \leq 1 - \frac{\delta}{q}$, with $\alpha_q(\delta) = 0$ for $(q-1)/q < \delta \leq 1$. We will be considering only binary codes, hence we have everywhere $q = 2$.

2.2 Bilingual and Trilingual Syntactic Codes

In this companion paper to [13], we carry out some analysis, based on the binary syntactic variables recorded in the SSWL database “Syntactic Structures of World Languages” [21]. We will not refer to these variables as “syntactic parameters”, because it is understood among linguists that the binary variables recorded in the SSWL database do not correspond to “syntactic parameters” the sense of the Principles and Parameters program, for example because of conflation of deep and surface structure. However, this database is useful because it contains a fairly large number of world languages (253) and of syntactic variables (115). From a computational perspective, one of the main problems in using SSWL data lies in the fact that not all 115 variables are mapped for all the 253 languages: indeed the languages are very non-uniformly mapped, with some (mostly Indo-European) languages mapped with 100% of the variables and others with only very few entries. Thus, for the purpose of the computations in this paper, when comparing a set of different languages, we have only used those variables that are fully mapped, in the SSWL database, for all the languages in the given set.

We consider here all the pairs $\{\ell_1, \ell_2\}$ of languages in the SSWL database, and all triples $\{\ell_1, \ell_2, \ell_3\}$, and the resulting codes $\mathcal{C}_{\ell_1, \ell_2}$ and $\mathcal{C}_{\ell_1, \ell_2, \ell_3}$. We refer to these as “bilingual and trilingual syntactic codes”. The computation grows rapidly much heavier for larger sets of languages. However, these cases are enough to see some interesting results on how the corresponding code parameters are distributed in the space of code parameters.

The fact that we consider sets of either two or three languages means that the corresponding codes have either two or three code words, namely the code parameter $k = \log_2(\#\mathcal{C})$ is fixed, with either $\#\mathcal{C} = 2$ or $\#\mathcal{C} = 3$.

As explained above, the length of the code words for these codes is not fixed: it is the largest number of syntactic binary variables in the SSWL list that are completely mapped for the languages in the set. Thus, for a set $\{\ell_1, \ell_2\}$ or $\{\ell_1, \ell_2, \ell_3\}$ we have a corresponding number of binary variables $n = n(\ell_1, \ell_2)$ or $n = n(\ell_1, \ell_2, \ell_3)$ with the codes $\mathcal{C}_{\ell_1, \ell_2} \subset \mathbb{F}_2^{n(\ell_1, \ell_2)}$ and $\mathcal{C}_{\ell_1, \ell_2, \ell_3} \subset \mathbb{F}_2^{n(\ell_1, \ell_2, \ell_3)}$. This is not a problem, since points in the space of code parameters correspond to codes of any arbitrary length. For each code $\mathcal{C}_{\ell_1, \ell_2}$ and $\mathcal{C}_{\ell_1, \ell_2, \ell_3}$ we compute the corresponding code parameters

$$(\delta(\mathcal{C}_{\ell_1, \ell_2}), R(\mathcal{C}_{\ell_1, \ell_2})) \quad \text{and} \quad (\delta(\mathcal{C}_{\ell_1, \ell_2, \ell_3}), R(\mathcal{C}_{\ell_1, \ell_2, \ell_3})),$$

and we plot them in the plane of code parameters. We then compare their position to different bounds in the space of code parameters.

Fig. 1 The code parameters of all 2 and 3 tuples of languages in the SSWL database. The horizontal and vertical axes represent the (δ, R) coordinates of the code parameters

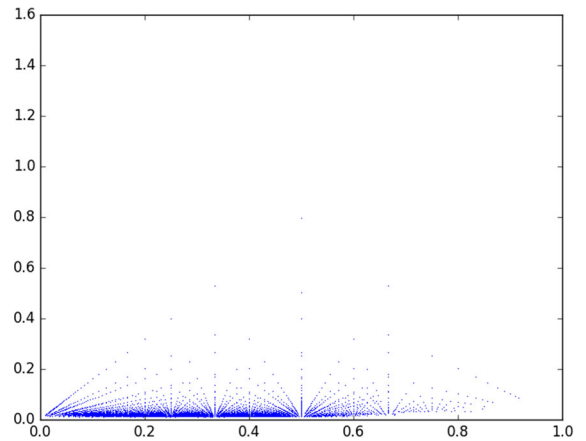
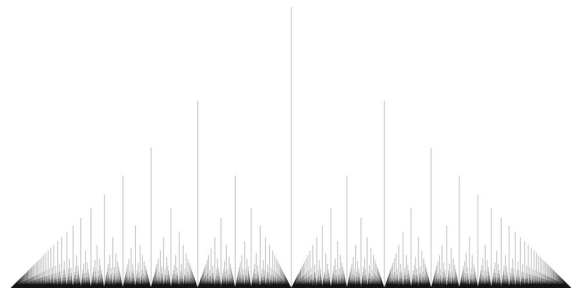


Fig. 2 The undergraph of the Thomae function in (δ, R) coordinates



3 Code Parameters of Syntactic Codes

The Python code is available at www.its.caltech.edu/~matilde/SSWLcodes. The `code_params.py` script has utilities that will compute the code parameters for a given subset of the languages. The plot for all the codes $\mathcal{C}_{\ell_1, \ell_2}$ and $\mathcal{C}_{\ell_1, \ell_2, \ell_3}$ of pairs and triples of languages in the SSWL database was generated with `fixed_sized_subset.py`. The resulting plot is given in Fig. 1. Note that for all the codes in this set the absolute rate of the code is always either $k(\mathcal{C}_{\ell_1, \ell_2}) = 1$ or $k(\mathcal{C}_{\ell_1, \ell_2, \ell_3}) = \log_2(3)$.

3.1 Thomae Function and Random Parameters

The fractal pattern that one sees appearing in Fig. 1 may seem at first surprising, but in fact it has a very simple explanation. Recall that the Thomae function defined as

$$f(x) = \begin{cases} 1 & x = 0; \\ \frac{1}{q} & x \in \mathbb{Q}, x = \frac{p}{q}, q > 0; \\ 0 & x \in \mathbb{R} \setminus \mathbb{Q}. \end{cases}$$

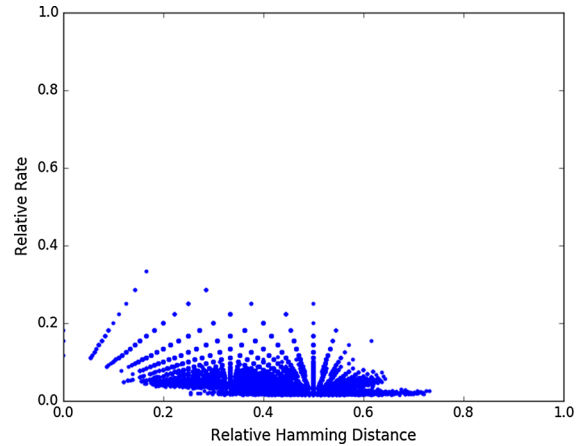
A plot of the undergraph of the Thomae function is shown in Fig. 2. One can clearly see the similarity with Fig. 1. Indeed, the reason why the code parameters in Fig. 1 approximate the Thomae function depends on the fact that we are fixing the absolute rate of the codes. We have code points

$$(\delta(\mathcal{C}), R(\mathcal{C})) = \left(\frac{d}{n}, k \cdot \frac{1}{n} \right).$$

Thus, since k is fixed to be either 1 or $\log_2(3)$, we obtain two copies, scaled by the respective values of k , of the graph of

$$f : \frac{d}{n} \mapsto \frac{1}{n},$$

Fig. 3 Plot of code points of random codes \mathcal{C} with $\#\mathcal{C} = 3$ drawn from a uniform random distribution



which, when $\gcd(d, n) = 1$, agrees with the Thomae function, restricted to those values of d and n that occur in our set of codes. We see that we obtain several additional points in Fig. 1 that lie in the undergraph of the Thomae function, which come from the cases with $\gcd(d, n) \neq 1$, where for $d = ur$ and $n = vr$, in addition to the plot point $(\frac{u}{v}, \frac{1}{v})$ on the graph of the Thomae function, we also find other lower plot points $(\frac{u}{v}, \frac{1}{vr})$ on the same vertical line.

We compare this behavior with the case of code parameters produced by randomly generated points. A plot of 3-tuples of a set of randomly generated parameters is shown in Fig. 3. These are random codes \mathcal{C} with $\#\mathcal{C} = 3$ (three code words) drawn from a uniform random distribution. This was generated with `random_parameters.py`. We see that the lower region of Fig. 1 behaves similarly to the case of randomly generated parameters. Indeed, it is known that the region in the space of code parameters that lies below the Gilbert–Varshamov line is typically the one that is populated by random codes [2,5].

3.2 Gilbert–Varshamov and Plotkin Bound

We want to analyze the position, with respect to various bounds, of the code parameters of codes $\mathcal{C}_{\mathcal{L}}$, for sets \mathcal{L} of languages and their SSWL syntactic data. In order to do that, since a systematic computation covering all cases for larger sets of languages lies beyond our available computational power, we check the position of code points on sets of randomly selected languages and syntactic parameters in the SSWL database. The script `random_subset.py` takes the list of binary syntactic variables recorded in the SSWL database and selects a random choice of a subset of these binary variables. Then it randomly selects a subset of languages, among those for which the selected set of parameters is completely mapped in the database.

In a typical plot obtained with this method, like the one shown in Fig. 4, we see curves that correspond to a superposition of several scaled Thomae functions, for varying values of k . A large part of the code points tends to cluster in the region below the Gilbert–Varshamov curve, similarly to what one would expect for random codes. However, there is typically a significant portion of the code points obtained in this way that lies above the Gilbert–Varshamov curve, populating the area between the Gilbert–Varshamov curve and the Plotkin line. Since these points tend to cover the entire width of the region between these two curves, which contains the asymptotic bound, certainly part of them will lie above the asymptotic bound, in the region of the sporadic codes. For some choices of random subsets of languages and parameters one also sees an even smaller number of code points that lie above the Plotkin bound (hence certainly above the asymptotic bound). This confirms the observation made in [13] regarding code points of syntactic codes and their position in the space of code parameters. Using randomized sets of SSWL parameters and corresponding sets of languages, we can also plot the density of code points in the various regions of the space of code parameters, as shown in Fig. 5.

Fig. 4 Code points for set of randomly selected SSWL syntactic variables and languages next to the Gilbert–Varshamov and the Plotkin bounds

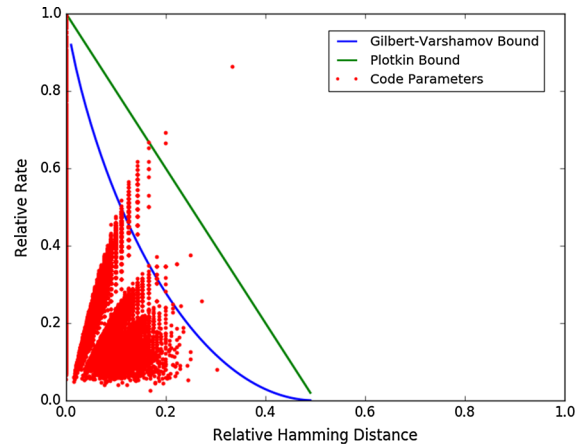
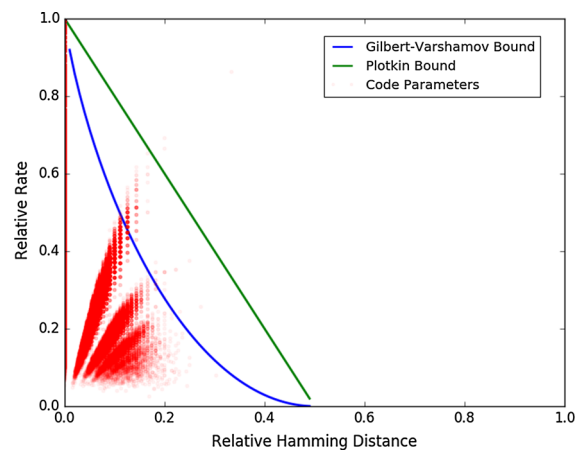


Fig. 5 Density of code points of syntactic codes, with respect to the Gilbert–Varshamov and the Plotkin bounds



4 Dynamics in the Space of Code Parameters

In [17] a dynamical model of language change was proposed, based on a spin glass model for syntactic parameters and language interactions. It is a simple model based on a graph with languages at the vertices, represented by a vector of their syntactic parameters interpreted as spin variables, and strengths of interaction between languages along the edges, measured using data proportional to the amount of bilingualism. In the case of syntactic parameters behaving as independent variables, in the low temperature regime (see [17] for a discussion of the interpretation of the temperature parameter in this model) the dynamics converges rapidly towards an equilibrium state where all the spin variables corresponding to a given syntactic feature for the various languages align to the value most prevalent in the initial configuration. The SSWL database does not record relations between parameters, although it can be shown by other approaches that interesting relations are present, see [14, 15]. Using syntactic data from [6, 7], which record explicit entailment relations between different parameters, it was shown in [17], for small graph examples, that in the presence of relations the dynamics settles on equilibrium states that are not necessarily given by completely aligned spins.

In this section we analyze how the dynamical system studied in [17] acts on the code parameters, with an initial condition given by the code obtained from the initial set of languages on which the dynamics of [17] acts. In the case without entailment relations, where each syntactic parameter evolves as an independent spin variable, each syntactic parameter (spin variable) tends to converge, in the low temperature regime, to an equilibrium configuration where all the languages in the set align their parameter to the value most prevalent in the initial condition. Thus, the dynamics (in this unrealistic case without relations between parameters) tends to make the languages more similar.

At the level of the codes, this will reduce the Hamming distance, while keeping the code rate unchanged, hence the dynamics tends to spoil the code. However, this is not necessarily the case if entailment relations are taken into account. We see cases where the dynamics can improve the code parameters. Moreover, while simulating the dynamics of the whole spin glass model of [17] with a large number of languages and entailment relations quickly becomes intractable, one can obtain information about the dynamics by studying its effect on the code parameters and by restricting to randomly chosen subsets of the original set of languages, to obtain a more tractable dynamical system. As we show below, one can obtain information about the convergence of the dynamics of the spin glass model by looking at the induced dynamics on the code parameters.

4.1 Spin Glass Models for Syntactic Parameters

When we interpret the dynamics of the model considered in [17] in terms of codes and the space of code parameters, the initial datum of the set of languages \mathcal{L} at the vertices of the graph, with its given list of syntactic binary variables, determines a code $\mathcal{C}_{\mathcal{L}}$. The absolute rate $k = k(\mathcal{C}_{\mathcal{L}}) = \log_2(\#\mathcal{C}_{\mathcal{L}})$ and the number of syntactic features considered $n = n(\mathcal{C}_{\mathcal{L}})$ remain fixed along the dynamics, hence the dynamics moves the code points along the horizontal lines with fixed R -coordinate. In the case of independent syntactic binary variables, the dynamics follows a gradient descent for an energy functional that is simply given by the Hamiltonian

$$H_{x_i} = - \sum_{\ell \neq \ell' \in \mathcal{L}} J_{\ell, \ell'} S_{x_i(\ell)} S_{x_i(\ell')},$$

for a given syntactic variable $x_i, i = 1, \dots, n$, where $J_{\ell, \ell'}$ is the strength of the interaction along the edge connecting the vertices ℓ and ℓ' and S_{x_i} is the ± 1 values spin variable associated to the binary variable x_i . The minimum of the energy H_{x_i} for the single variable x_i is achieved when $S_{x_i(\ell)} S_{x_i(\ell')} = 1$, that is, when $|x_i(\ell) - x_i(\ell')| = 0$. Thus, in this case where each syntactic variable runs as an independent Ising model, the minimum is achieved where $d_H(x(\ell), x(\ell')) = \sum_{i=1}^n |x_i(\ell) - x_i(\ell')| = 0$, that is, when all the spins align. In the presence of entailment relations between different syntactic variables, it was shown in [17] that the Hamiltonian should be modified by a term that introduces the relations as a Lagrange multiplier. This alters the dynamics and the equilibrium state, depending on a parameter that measures how strongly enforced the relations are. From the point of view of coding theory discussed here, it seems more reasonable to modify this dynamical system, so that it can be better described as a dynamics in the space of code parameters. It is natural therefore to consider a similar setting, where we assign a given set \mathcal{L} of languages to the vertices of a complete graph $G_{\mathcal{L}}$, with assigned energies $J_e = J_{\ell, \ell'}$ at the edges $e \in E(G_{\mathcal{L}})$ with $\partial e = \{\ell, \ell'\}$. We denote by $x(\ell) = (x_j(\ell))_{j=1}^n$ the vector of binary variables that lists the n syntactic features of the language ℓ . We consider these as maps $x : \mathcal{L} \rightarrow \{0, 1\}^n$, or equivalently as points $x \in \{0, 1\}^{n \cdot \mathcal{L}}$. Consider an energy functional of the form

$$H(x) = \sum_{\ell \neq \ell' \in \mathcal{L}} J_{\ell, \ell'} d_H(x(\ell), x(\ell')),$$

for $J_{\ell, \ell'} = J_e > 0$, where $d_H(x(\ell), x(\ell'))$ is the Hamming distance,

$$H(x) = \sum_{e \in E(G_{\mathcal{L}})} \sum_{j=1}^n J_e |x_j(\ell) - x_j(\ell')|.$$

The corresponding partition function is given by

$$Z = \sum_{x \in \{0, 1\}^{n \cdot \mathcal{L}}} e^{-\beta H(x)}.$$

At low temperature (large β), the partition function is concentrated around the minimum of $H(x)$, that is, were all $d_H(x(\ell), x(\ell')) = 0$, hence where all the vectors $x(\ell) \in \{0, 1\}^n$ agree. Given an initial condition $x^0 \in \{0, 1\}^{n \cdot \mathcal{L}}$, with $x^0(\ell) \in \{0, 1\}^n$ for $\ell \in \mathcal{L}$, and the datum $(J_e)_{e \in E(G_{\mathcal{L}})}$ of the strengths of the interaction energies along

the edges, the same method used in [17], based on the standard Metropolis–Hastings algorithm, can be used to study the dynamics in this setting, with a similar behavior. In the space of code parameters, given the code point $(\delta^0, R^0) = (\delta(\mathcal{C}(x^0)), R(\mathcal{C}(x^0)))$ associated to the initial condition x^0 , the dynamics moves the code point along the line with constant $R = R^0$. As the dynamics approaches the minimum of the action, the code point enters the region below the Gilbert–Varshamov bound, as it moves towards smaller values of δ .

4.2 Dynamics in the Presence of Entailment Relations

It is more interesting to see what happens in the case of where the syntactic variables are not independent but involve entailment relations between syntactic parameters. To this purpose we need to use syntactic data from [6, 7], where relations between syntactic parameters are explicitly recorded. A typical form of the relations described in [6, 7] consists of two syntactic parameters x_i and x_j , where x_i is unconstrained and has binary values $x_i \in \{0, 1\}$, while the values of x_j are constrained by the value of x_i , so that if $x_i = 1$ x_j can take any of two binary values, while if $x_i = 0$ then x_j becomes undefined. We express this by considering x_j as a ternary valued variables, $x_j \in \{-1, 0, +1\} \simeq \mathbb{F}_3$, where $x_j = \pm 1$ stand for the ordinary binary values and $x_j = 0$ signifies undefined. We can then write the relation in the form of a function

$$\mathcal{R}_{ij}(x) = |x_i - |x_j||,$$

where the solutions to $\mathcal{R}_{ij}(x) = 0$ are precisely $(x_i = 1, x_j = \pm 1)$ and $(x_i = 0, x_j = 0)$. We introduce a parameter $E_{ij} \geq 0$ that measures how strongly the relation \mathcal{R}_{ij} is enforced. The modified energy functional that accounts for the presence of a relation between the i -th and the j -th parameter is then of the form

$$H(x) = \sum_{\ell \neq \ell' \in \mathcal{L}} J_{\ell, \ell'} d_H(x(\ell), x(\ell')) + \sum_{\ell} E_{ij}(\ell) \mathcal{R}_{ij}(x(\ell)),$$

where we allow here for the possibility that the relation may be differently (more strongly or weakly) enforces for different $\ell \in \mathcal{L}$. In practice, it will be convenient to assume that $E_{ij}(\ell) = E_{ij}$ is independent of $\ell \in \mathcal{L}$. When considering all the possible relations between different parameters in the list, that are of the form \mathcal{R}_{ij} as above, we separate out the set $\{1, \dots, n\}$ of all the syntactic parameters in the list in two sets, $\{1, \dots, n\} = B \cup T$, where B is the set of independent binary variables and T is the set of entailed ternary variables, and we write the energy functional as

$$H(x) = \sum_{\ell \neq \ell' \in \mathcal{L}} J_{\ell, \ell'} d_H(x(\ell), x(\ell')) + \sum_{i \in B, j \in T} E_{ij} \mathcal{R}_{ij}(x),$$

where

$$\mathcal{R}_{ij}(x) = \sum_{\ell \in \mathcal{L}} \mathcal{R}_{ij}(x(\ell))$$

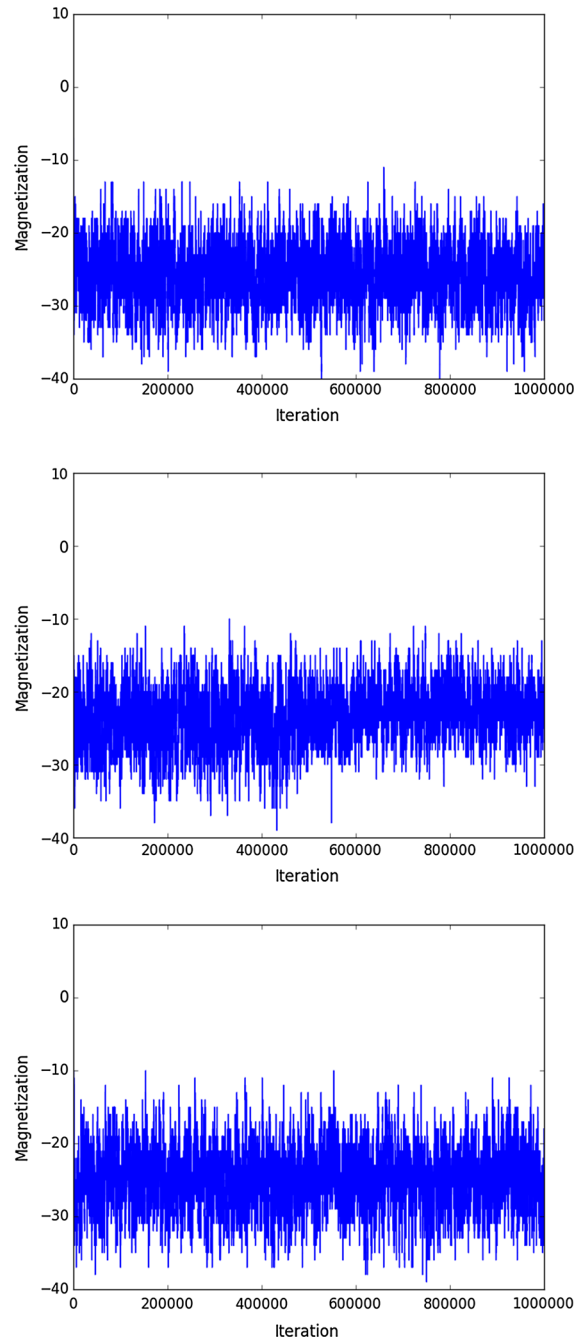
and where $E_{ij} = 0$ if there is no direct dependence of x_j upon x_i . One can consider additional terms $\mathcal{R}_{i_1, \dots, i_r, j}(x)$ with $i_a \in B, j \in T$ and $j \in T$ of a similar form, when the ternary parameter x_j is entailed by more than one binary parameter x_i , and add them to the energy functional in a similar way, with entailment energies $E_{i_1, \dots, i_r, j} \geq 0$.

4.3 Dynamics in the Space of Code Parameters

In these models, where part of the syntactic variables $x_i \in B$ are seen as binary variables and part $x_j \in T$ as ternary variables, for the purpose of coding theory, we consider the whole $x = (x_i)_{i=1}^n$ as a vector in \mathbb{F}_3^n , in order to compute the code parameters of the resulting code $\mathcal{C}(\mathcal{L}) \subset \mathbb{F}_3^n$.

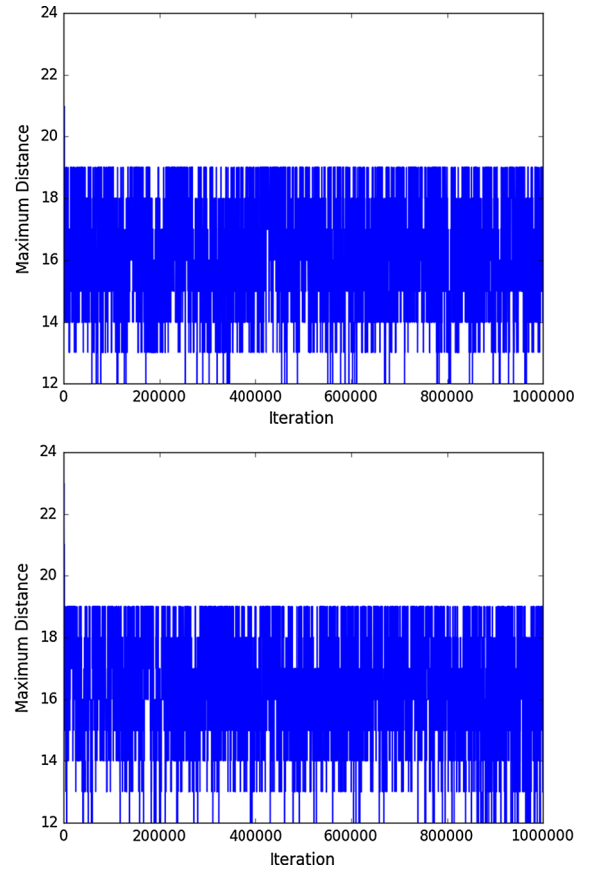
One can see already in a very simple example, and using the dynamical system in the form described in [17], that the dynamics in the space of code parameters now does not need to move towards the $\delta = 0$ line. Consider the very

Fig. 6 Average magnetization for the spin glass model of [17] computed for the languages and parameters of [6], in the cases with $T = 10$ and $E = 0$; $T = 10$ and $E = 9000$; $T = 910$ and $E = 0$



small example, with just two entailed syntactic variables and four languages, discussed in [17], where the chosen languages are $\mathcal{L} = \{\ell_1, \ell_2, \ell_3, \ell_4\} = \{\text{English, Welsh, Russian, Bulgarian}\}$ and the two syntactic parameters are $\{x_1, x_2\} = \{\text{StrongDeixis, StrongAnaphoricity}\}$. Since we have an entailment relation, the possible values of the variables x_i are now ternary, $x_i(\ell) \in \{0, -1, +1\}$, that is, we consider here codes $\mathcal{C} \subset \mathbb{F}_3^n$. In this example $n = 2$. The initial condition x^0 is given by

Fig. 7 Dynamics in the space of code parameters: maximal distance



$$\begin{aligned}
 x^0(\ell_1) &= (+1, +1) \\
 x^0(\ell_2) &= (-1, 0) \\
 x^0(\ell_3) &= (+1, +1) \\
 x^0(\ell_4) &= (+1, +1).
 \end{aligned}$$

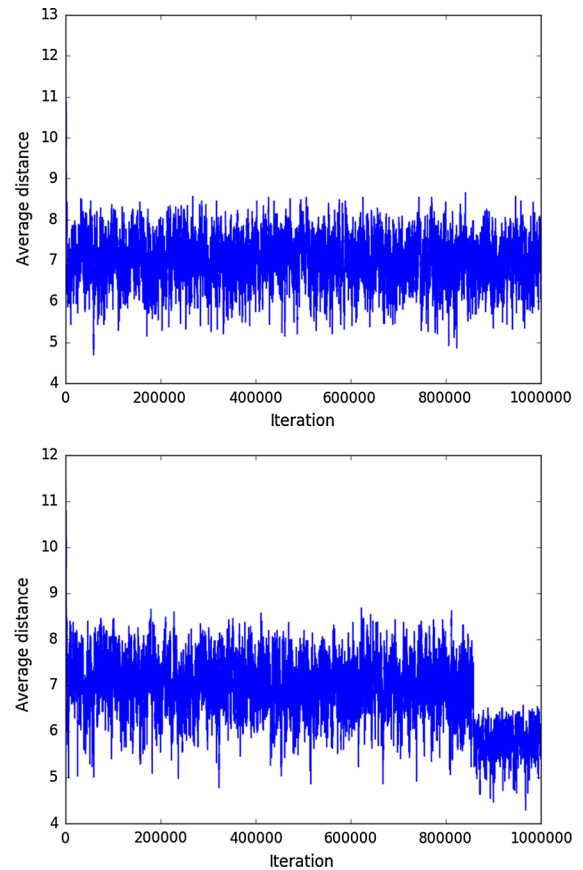
Note that, since we have two identical code words $x^0(\ell_1) = x^0(\ell_3)$ in this initial condition, the parameter $d(\mathcal{C}_{\mathcal{L}}) = 0$, so the code point $(\delta(\mathcal{C}_{\mathcal{L}}), R(\mathcal{C}_{\mathcal{L}})) = (0, \log_3(2))$ already lies on the vertical line $\delta = 0$. We consider in this case the same dynamical system used in [17] to model the case with entailment, which is a modification of the Ising model to a coupling of an Ising and a Potts model with $q = 3$ at the vertices of the graph. This dynamics, which depends on the temperature parameter $T = 1/\beta$ and on an auxiliary parameter E , the “entailment energy”, that measures how strongly the entailment relation is enforced. In the cases with high temperature and either high or low entailment energy, it is shown in [17] that one can have equilibrium states like

$$\begin{aligned}
 x(\ell_1) &= (+1, 0) \\
 x(\ell_2) &= (+1, -1) \\
 x(\ell_3) &= (-1, 0) \\
 x(\ell_4) &= (+1, +1),
 \end{aligned}$$

for the high entailment energy case, or

$$\begin{aligned}
 x(\ell_1) &= (+1, -1) \\
 x(\ell_2) &= (-1, -1) \\
 x(\ell_3) &= (-1, +1) \\
 x(\ell_4) &= (-1, -1),
 \end{aligned}$$

Fig. 8 Dynamics in the space of code parameters: average distance



for the low entailment energy case. In both of these cases, the minimum distance $d = \min_{\ell \neq \ell'} d_H(x(\ell), x(\ell')) = 1$, hence $\delta = 1/2$. Thus, along the dynamics, the code point in the space of code parameters has moved away from the line $\delta = 0$, along the line with constant R . The final code point with $\delta = 1/2$ and $R = \log_3(2)$ lies above the GV-curve $R = 1 - H_3(\delta)$. Thus, in this very simple example we have seen that the dynamics in the case where syntactic parameters are not independent variables can in fact move the code toward a better code, passing from below to above the GV-bound.

4.4 Simulations

The example mentioned above is too simple and artificial to be significant, but we can analyze a more general situation, where we consider the full syntactic data of [6,7], with all the entailment relations taken into account, and the same interaction energies along the edges as in [17], taken from the data of [16], which can be regarded as roughly proportional to a measure of the amount of bilingualism.

When we work with the full set of data from [6,7], involving 63 parameters for 28 languages (from which we exclude 4 languages that do not occur in the [16] data), we see that the large size of the graph and the presence of many entailment relations render the dynamics a lot more complicated than the simple examples discussed in [17]. Indeed for such a large graph the convergence of the dynamics becomes extremely slow, even in the low temperature case and even when entailment relations are switched off, as shown in the graph of the average magnetization in Fig. 6. Such a large system becomes computationally too heavy, and it is difficult to handle a sufficiently large iterations to get to see any convergence effect. However, when one considers codes obtained by extracting arbitrary subsets of three languages from this set and follows them along the dynamics, computing the corresponding position

in the space of code parameters, one sees that, in the case without entailment ($E = 0$) the average distance drops notably after enough iterations, as shown in Figs. 7 and 8 indicating that the simulation might in fact converge, even though at the same state in the number of iteration the average magnetization is not settling yet. In the case with entailment relations one should expect the convergence process to be even slower. Moreover, as in the small example discussed above, the δ parameter may settle on a limit value different than zero, so the data of the simulation are less informative.

References

1. Baker, M.: *The Atoms of Language*. Basic Books, New York (2001)
2. Barg, A., Forney, G.D.: Random codes: minimum distances and error exponents. *IEEE Trans. Inf. Theory* **48**(9), 2568–2573 (2002)
3. Chomsky, N.: *Lectures on Government and Binding*. Foris Publications, Dordrecht (1982)
4. Chomsky, N., Lasnik, H.: The theory of Principles and Parameters, in “Syntax: An international handbook of contemporary research”, pp. 506–569, de Gruyter (1993)
5. Coffey, J.T., Goodman, R.M.: Any code of which we cannot think is good. *IEEE Trans. Inf. Theory* **36**(6), 1453–1461 (1990)
6. Longobardi, G., Guardiano, C.: Evidence for syntax as a signal of historical relatedness. *Lingua* **119**, 1679–1706 (2009)
7. Longobardi, G., Guardiano, C., Silvestri, G., Boattini, A., Ceolin, A.: Towards a syntactic phylogeny of modern Indo-European languages. *J. Hist. Linguist.* **3**(1), 122–152 (2013)
8. Manin, Y.I.: What is the maximum number of points on a curve over \mathbb{F}_2 ? *J. Fac. Sci. Univ. Tokyo Sect. IA Math* **2**(83), 715–720 (1982)
9. Manin, Y.I.: A computability challenge: asymptotic bounds and isolated error-correcting codes, in “Computation, physics and beyond”, pp. 174–182, *Lecture Notes in Comput. Sci.*, vol. 7160, Springer (2012)
10. Manin, Y.I.: Complexity vs Energy: Theory of Computation and Theoretical Physics, [arXiv:1302.6695](https://arxiv.org/abs/1302.6695) [cs.CC]
11. Manin, Y.I., Marcolli, M.: Error-correcting codes and phase transitions. *Math. Comput. Sci.* **5**, 133–170 (2011)
12. Manin, YuI, Marcolli, M.: Kolmogorov complexity and the asymptotic bound for error-correcting codes. *J. Differ. Geom.* **97**, 91–108 (2014)
13. Marcolli, M.: Syntactic parameters and a coding theory perspective on entropy and complexity of language families. *Entropy* **18**(4), 110 (2016)
14. Park, J.J., Boettcher, R., Zhao, A., Mun, A., Yuh, K., Kumar, V., Marcolli, M.: Prevalence and recoverability of syntactic parameters in sparse distributed memories, [arXiv:1510.06342](https://arxiv.org/abs/1510.06342) [cs.CL]
15. Port, A., Gheorghita, I., Guth, D., Clark, J.M., Liang, C., Dasu, S., Marcolli, M.: Persistent Topology of Syntax, [arXiv:1507.05134](https://arxiv.org/abs/1507.05134) [cs.CL]
16. Ronen, S., Goncalves, B., Hu, K.Z., Vespignani, A., Pinker, S., Hidalgo, C.A.: Links that speak: the global language network and its association with global fame. *Proc. Natl. Acad. Sci.* **111**(52), E5616–E5622 (2014)
17. Siva, K., Tao, J., Marcolli, M.: Spin Glass Models of Syntax and Language Evolution, [arXiv:1508.00504](https://arxiv.org/abs/1508.00504) [cs.CL]
18. Tsfasman, M.A., Vladut, S.G.: *Algebraic-geometric codes, Mathematics and its Applications (Soviet Series)*, vol. 58, Kluwer Academic Publishers (1991)
19. Tsfasman, M.A., Vladut, S.G., Zink, T.: Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound. *Math. Nachr.* **109**, 21–28 (1982)
20. Vladut, S.G., Drinfeld, V.G.: The number of points of an algebraic curve. *Funktsional. Anal. I Prilozhen.* **17**(1), 68–69 (1983)
21. SSWL Database of Syntactic Parameters: <http://sswl.railsplayground.net/>