Towards a Geometry of Syntax

Matilde Marcolli
MAT1509HS: Mathematical and Computational Linguistics

University of Toronto, Winter 2019, T 4-6 and W 4, BA6180 this lecture based on:

- Matilde Marcolli, *Syntactic parameters and a coding theory perspective on entropy and complexity of language families*, Entropy 18 (2016), no. 4, Paper No. 110, 17 pp.
- Kevin Shu and Matilde Marcolli, *Syntactic structures and code parameters*, Mathematics in Computer Science 11 (2017) no. 1, 79–90.
- J.J. Park, R. Boettcher, A. Zhao, A. Mun, K. Yuh, V. Kumar, M. Marcolli, *Prevalence and recoverability of syntactic parameters in sparse distributed memories*, in "Geometric Science of Information. Third International Conference GSI 2017", pp. 265–272, Lecture Notes in Computer Science, Vol.10589, Springer 2017.
- Andrew Ortegaray, Robert C. Berwick, Matilde Marcolli, *Heat Kernel Analysis of Syntactic Structures*, arXiv:1803.09832

What kind of relations exist between syntactic parameters?

- Entailment relations: some explicitly known relations where one state of a parameter (or more) can make another parameter undefined
- Example: $\{p_1, p_2\} = \{\text{Strong Deixis}, \text{Strong Anaphoricity}\}$

	p_1	p_2
ℓ_1	$+1$	$+1$
ℓ_2	-1	0
ℓ_3	$+1$	$+1$
ℓ_4	$+1$	-1

$\{\ell_1, \ell_2, \ell_3, \ell_4\} = \{\text{English}, \text{Welsh}, \text{Russian}, \text{Bulgarian}\}$

- several entailment relations are recorded in the data of Longobardi–Guardiano

- SSWL database does not record relations between parameters
- relations can be detected through methods of data analysis
- goals: identify a good set of independent variables among syntactic parameters, understand (at least statistically) the "manifold" determined by the relations
- some methods we consider here:

1. coding theory: code parameters, position in the space of codes
2. Kanerva networks: sparse distributed memories
3. heat kernel dimensional reduction: Laplace eigenfunctions

Coding Theory to study how syntactic structures differ across the landscape of human languages

• Kevin Shu, Matilde Marcolli, *Syntactic Structures and Code Parameters*, arXiv:1610.00311

• Matilde Marcolli, *Syntactic Parameters and a Coding Theory Perspective on Entropy and Complexity of Language Families*, Entropy 2016, 18(4), 110

- select a group of languages $\mathcal{L} = \{\ell_1, \ldots, \ell_N\}$
- with the binary strings of n syntactic parameters form a code $\mathcal{C}(\mathcal{L}) \subset \mathbb{F}_2^n$
- compute code parameters $(R(\mathcal{C}), \delta(\mathcal{C}))$ code rate and relative minimum distance
- analyze position of (R, δ) in space of code parameters
- get information about "syntactic complexity" of $\mathcal{L}$

Error-correcting codes

- *Alphabet*: finite set A with $\#A = q \geq 2$.
- *Code*: subset $C \subset A^n$, *length* $n = n(C) \geq 1$.
- *Code words*: elements $x = (a_1, \ldots, a_n) \in C$.
- *Code language*: $\mathcal{W}_C = \cup_{m \geq 1} \mathcal{W}_{C,m}$, words $w = x_1, \ldots, x_m$; $x_i \in C$.
- *ω-language*: Λ_C, infinite words $w = x_1, \ldots, x_m, \ldots$; $x_i \in C$.
- Special case: $A = \mathbb{F}_q$, *linear codes*: $C \subset \mathbb{F}_q^n$ linear subspace
- in general: *unstructured codes*

Code parameters

- $k = k(C) := \log_q \#C$ and $[k] = [k(C)]$ integer part of $k(C)$

$$q^{[k]} \leq \#C = q^k < q^{[k]+1}$$

- *Hamming distance*: $x = (a_i)$ and $y = (b_i)$ in C

$$d((a_i),(b_i)) := \#\{i \in (1,\dots,n) \,|\, a_i \neq b_i\}$$

- *Minimal distance* $d = d(C)$ of the code

$$d(C) := \min \{d(a,b) \,|\, a,b \in C, a \neq b\}$$

Codes and code parameters: binary codes

error correcting codes $\mathcal{C} \subset \mathbb{F}_2^n$

• transmission rate (encoding)

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

for q-ary codes in $\mathbb{F}_q^n$ take $k = \log_q(N)$

• relative minimum distance (decoding)

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

Hamming distance of binary strings of ℓ_1 and ℓ_2

• error correcting codes: optimize for maximal R and δ but constraints that make them inversely correlated

• bounds in the space of code parameters (R, δ)

The space of code parameters:

• $Codes_q$ = set of all codes C on an alphabet $\#A = q$

• function $cp : Codes_q \to [0,1]^2 \cap \mathbb{Q}^2$ to code parameters
$cp : C \mapsto (R(C), \delta(C))$

• the function $C \mapsto (R(C), \delta(C))$ is a *total recursive map* (Turing computable)

• *Multiplicity* of a code point (R, δ) is $\#cp^{-1}(R, \delta)$

- M.A. Tsfasman, S.G. Vladut, *Algebraic-geometric codes*, Mathematics and its Applications (Soviet Series), Vol. 58, Kluwer Academic Publishers, 1991.

Bounds on code parameters

- singleton bound: $R + \delta \leq 1$
- Gilbert-Varshamov curve (q-ary codes)

$$R = 1 - H_q(\delta), \quad H_q(\delta) = \delta \log_q (q-1) - \delta \log_q \delta - (1-\delta) \log_q (1-\delta)$$

q-ary Shannon entropy: asymptotic behavior of volumes of Hamming balls for large n

- The Gilbert-Varshamov curve represents the typical behavior of large random codes (Shannon Random Code Ensemble)
- Note: if syntactic parameters really were identically distributed independent random variables, subject to an evolution via a Markov model on a tree (simple assumption of phylogenetic models) then would expect codes from sets of languages to behave like Shannon random codes
- distance from SRCE behavior measures presence of relations that affect distribution of syntactic parameters across languages

Statistics of codes and the Gilbert–Varshamov bound

Known *statistical* approach to the GV bound: *random codes*

Shannon Random Code Ensemble: ω-language with alphabet A; uniform Bernoulli measure on Λ_A; choose code words of C as independent random variables in this measure

Volume estimate:

$$q^{(H_q(\delta)-o(1))n} \leq Vol_q(n, d = n\delta) = \sum_{j=0}^{d} \binom{n}{j} (q-1)^j \leq q^{H_q(\delta)n}$$

Gives probability of parameter δ for SRCE meets the GV bound with probability exponentially (in n) near 1: expectation

$$\mathbb{E} \sim \binom{q^k}{2} Vol_q(n, d) q^{-n} \sim q^{n(H_q(\delta)-1+2R)+o(n)}$$

• typical random codes populate the region of code parameters below the Gilbert–Varshamov curve

The asymptotic bound

• Yu.I. Manin, *What is the maximum number of points on a curve over $\mathbb{F}_2$?* J. Fac. Sci. Tokyo, IA, Vol. 28 (1981), 715–720.

• existence proved by spoiling operations on codes

• separates space $[0,1]^2$ of code parameters into region below asymptotic bound $R = \alpha_q(\delta)$ where code points dense and with infinite multiplicity from region above where code points isolated and with finite multiplicity

• the function $R = \alpha_q(\delta)$ may be non-computable, but only as bad as Kolmogorov complexity (becomes computable given an oracle that orders codes by their Kolmogorov complexity)

- Yu.I. Manin, M. Marcolli, *Error-correcting codes and phase transitions*, Mathematics in Computer Science, Vol.5 (2011) 133–170
- Yu.I. Manin, M. Marcolli, *Kolmogorov complexity and the asymptotic bound for error-correcting codes*, Journal of Differential Geometry, Vol.97 (2014) 91–108

Spoiling operations on codes: C an $[n,k,d]_q$ code

- $C_1 := C *_i f \subset A^{n+1}$

$$(a_1,\dots,a_{n+1}) \in C_1 \text{ iff } (a_1,\dots,a_{i-1},a_{i+1},\dots,a_{n+1}) \in C\,,$$

and $a_i = f(a_1,\dots,a_{i-1},a_{i+1}\dots,a_{n+1})$
C_1 an $[n+1,k,d]_q$ code (f constant function)

- $C_2 := C*_i \subset A^{n-1}$

$(a_1,\dots,a_{n-1}) \in C_2$ iff $\exists b \in A,\ (a_1,\dots,a_{i-1},b,a_{i+1},\dots,a_{n-1}) \in C.$

C_2 an $[n-1,k,d]_q$ code

- $C_3 := C(a,i) \subset C \subset A^n$

$$(a_1,\dots,a_n) \in C_3 \text{ iff } a_i = a.$$

C_3 an $[n-1, k-1 \le k' < k, d' \ge d]_q$ code

Asymptotic bound

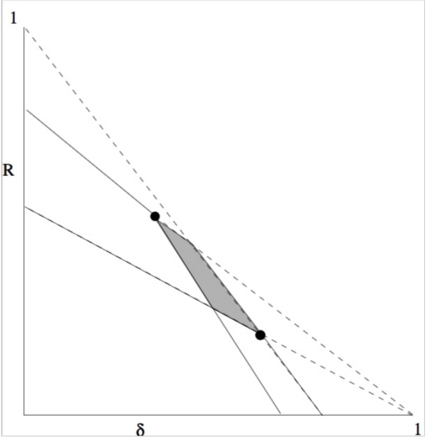
- Yu.I.Manin, *What is the maximum number of points on a curve over $\mathbb{F}_2$?* J. Fac. Sci. Tokyo, IA, Vol. 28 (1981), 715–720.

- $V_q \subset [0,1]^2$: all code points $(R,\delta) = cp(C)$, $C \in Codes_q$
- U_q: set of limit points of V_q
- Asymptotic bound: U_q all points below graph of a function

$$U_q = \{(R,\delta) \in [0,1]^2 \,|\, R \leq \alpha_q(\delta)\}$$

- Isolated code points: $V_q \smallsetminus (V_q \cap U_q)$

Method: controlling quadrangles



$R = \alpha_q(\delta)$ continuous decreasing function with $\alpha_q(0) = 1$ and $\alpha_q(\delta) = 0$ for $\delta \in [\frac{q-1}{q}, 1]$; has inverse function on $[0, (q-1)/q]$; U_q union of all lower cones of points in $\Gamma_q = \{R = \alpha_q(\delta)\}$

Characterization of the asymptotic bound

• Code points and multiplicities

• Set of code points of infinite multiplicity
$U_q \cap V_q = \{(R,\delta) \in [0,1]^2 \cap \mathbb{Q}^2 \mid R \leq \alpha_q(\delta)\}$ below the asymptotic bound

• Code points of finite multiplicity all above the asymptotic bound
$V_q \smallsetminus (U_q \cap V_q)$ and isolated (open neighborhood containing (R,δ) as unique code point)

Questions:

• Is there a characterization of the isolated good codes on or above the asymptotic bound?

Estimates on the asymptotic bound

- Plotkin bound:

$$\alpha_q(\delta) = 0, \quad \delta \geq \frac{q-1}{q}$$

- singleton bound:

$$\alpha_q(\delta) \leq 1 - \delta$$

- Hamming bound:

$$\alpha_q(\delta) \leq 1 - H_q(\frac{\delta}{2})$$

- Gilbert–Varshamov bound:

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

- difficult to construct codes above the asymptotic bound: examples from algebro-geometric codes from curves (but only for $q \geq 49$ otherwise entirely below the GV curve)

Computability question

• Note: only the asymptotic bound marks a significant change of behavior of codes across the curve (isolated and finite multiplicity/accumulation points and infinite multiplicity)

• in this sense it is very different from all the other bounds in the space of code parameters

• .... but no explicit expression for the curve $R = \alpha_q(\delta)$

• ... is the function $R = \alpha_q(\delta)$ computable?

• ... a priori no good statistical description of the asymptotic bound: is there something replacing Shannon entropy characterizing Gilbert–Varshamov curve?

- Yu.I. Manin, *A computability challenge: asymptotic bounds and isolated error-correcting codes*, arXiv:1107.4246

The asymptotic bound and Kolmogorov complexity

• while random codes are related to Shannon entropy (through the GV-bound) good codes and the asymptotic bound are related to Kolmogorov complexity

• the asymptotoc bound $R = \alpha_q(\delta)$ becomes computable given an oracle that can list codes by increasing Kolmogorov complexity

• given such an oracle: iterative (algorithmic) procedure for constructing the asymptotic bound

• ... it is at worst as "non-computable" as Kolmogorov complexity

• asymptotic bound can be realized as phase transition curve of a statistical mechanical system based on Kolmogorov complexity

- Yu.I. Manin, M. Marcolli, *Kolmogorov complexity and the asymptotic bound for error-correcting codes*, Journal of Differential Geometry, Vol.97 (2014) 91–108

Complexity

- How does one measure complexity of a physical system?

- Kolmogorov complexity: measures length of a minimal algorithmic description

... but ... gives very high complexity to completely random things

- Shannon entropy: measures average number of bits, for objects drawn from a statistical ensemble

- There are other proposals for complexity, but more difficult for formulate

- Gell-Mann complexity: complexity is high in an intermediate region between total order and complete randomness

Kolmogorov complexity

- Let $T_{\mathcal{U}}$ be a universal Turing machine (a Turing machine that can simulate any other arbitrary Turing machine: reads on tape both the input and the description of the Turing machine it should simulate)
- Given a string w in an alphabet $\mathfrak{A}$, the Kolmogorov complexity

$$\mathcal{K}_{T_{\mathcal{U}}}(w) = \min_{P:\, T_{\mathcal{U}}(P)=w} \ell(P),$$

minimal length of a program that outputs w

- universality: given any other Turing machine T

$$\mathcal{K}_T(w) = \mathcal{K}_{T_{\mathcal{U}}}(w) + c_T$$

shift by a bounded constant, independent of w; c_T is the Kolmogorov complexity of the program needed to describe T for $T_{\mathcal{U}}$ to simulate it

- any program that produces a description of w is an upper bound on Kolmogorov complexity $\mathcal{K}_{T_{\mathcal{U}}}(w)$
- think of Kolmogorov complexity in terms of data compression
- shortest description of w is also its most compressed form
- can obtain upper bounds on Kolmogorov complexity using data compression algorithms
- finding upper bounds is easy... but NOT lower bounds

Main problem

Kolmogorov complexity is NOT a computable function

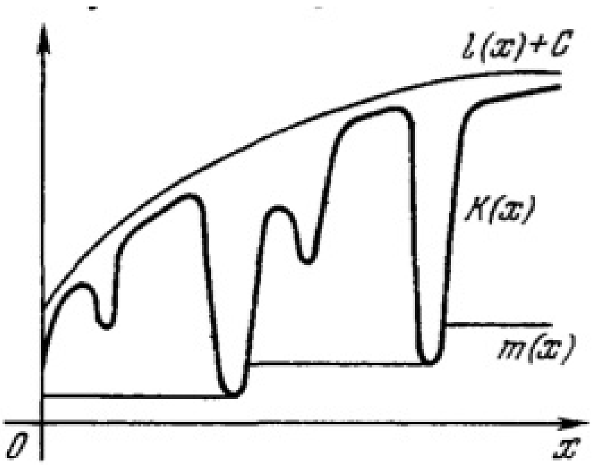
• suppose list programs P_k (increasing lengths) and run through $T_{\mathcal{U}}$: if machine halts on P_k with output w then $\ell(P_k)$ is an upper bound on $\mathcal{K}_{T_{\mathcal{U}}}(w)$

• but... there can be an earlier P_j in the list such that $T_{\mathcal{U}}$ has not yet halted on P_j

• if eventually halts and outputs w then $\ell(P_j)$ is a better approximation to $\mathcal{K}_{T_{\mathcal{U}}}(w)$

• would be able to compute $\mathcal{K}_{T_{\mathcal{U}}}(w)$ if can tell exactly on which programs P_k the machine $T_{\mathcal{U}}$ halts

• but... halting problem is unsolvable



with $m(x) = \min_{y \geq x} \mathcal{K}(y)$

Main Idea:

• use characterization of asymptotic bound as separating code points with finite multiplicity from code points with infinite multiplicity

• given the function from codes to code parameter, want an algorithmic procedure that inductively constructs preimage sets with finite/infinite multiplicity

• choose an ordering of code points: at step m list code points in order up to some growing size N_m

• initialize A_1: a set of *a preimage* for each code point up to N_1; initialize $B_1 = \emptyset$

• want to increase at each step A_m and B_m so that the first set only contains code points with multiplicity m

• going from step m to step $m+1$: new code points listed between N_m and N_{m+1} are added to A_m, and then points (previously in A_m or added) that do not have an $m+1$-st preimage are moved to B_{m+1}

• as $m \to \infty$ the sets A_m converge to set of code points of infinite multiplicity and the B_m converge to set of code points of finite multiplicity

• key problem: need to search for the $m+1$-st preimage to detect if a code point stays in A_{m+1} or is moved to B_{m+1}

• ordinarily this would involve an *infinite search*...

• ordering and complexity: use a relation between ordering and complexity that shows that only need to search among bounded complexity codes, so a *complexity oracle* will render the search finite

Conclusion: if asymptotic bound non-computable, only as bad as Kolmogorov complexity

Application to Linguistics: Syntactic Parameters and Coding

- M. Marcolli, *Principles and Parameters: a coding theory perspective*, arXiv:1407.7169

• idea: assign a (binary or ternary) code to a family of languages and use position of code parameters with respect to the asymptotic bound to test relatedness

• N = number of syntactic parameters $\Pi = (\Pi_\ell)_{\ell=1}^N$
each Π_ℓ with values in $\mathbb{F}_2 = \{0, 1\}$
(or $\mathbb{F}_3 = \{-1, 0, +1\}$ if include parameters that are not set in certain languages)

• $\mathcal{F} = \{L_k\}_{k=1}^m$ a set of natural languages (language "family")

• Code $C = C(\mathcal{F})$ in $\mathbb{F}^N$ ($\mathbb{F}_2^N$ or $\mathbb{F}_3^N$) with m code words $w_k = \Pi(L_k)$ string of syntactic parameters for the language L_k

Interpretation of Code Parameters

• $R = R(C)$ measures ratio between logarithmic size of number of languages in $\mathcal{F}$ and total number of parameters: how $\mathcal{F}$ distributed in the ambient $\mathbb{F}^N$

• $\delta = \delta(C)$ is the minimum, over all pairs of languages L_i, L_j in $\mathcal{F}$ of the relative Hamming distance

$$\delta(C(\mathcal{F})) = \min_{L_i \neq L_j \in \mathcal{F}} \delta_H(L_i, L_j)$$

$$\delta_H(L_i, L_j) = \frac{1}{N} \sum_{\ell=1}^{N} |\Pi_\ell(L_i) - \Pi_\ell(L_j)|$$

• code parameter δ used in Parameter Comparison Method for reconstruction of phylogenetic trees

Interpretation of Spoiling Operations

- first spoiling operation: effect of including one syntactic parameter in the list which is dependent on the other parameters
- second spoiling operation: forgetting one of the syntactic parameters
- third spoiling operation: forming subfamilies by considering languages that have a common value of one of the parameters

Parameters from Modularized Global Parameterization Method

- G. Longobardi, *Methods in parametric linguistics and cognitive history*, Linguistic Variation Yearbook, Vol.3 (2003) 101–138
- G. Longobardi, C. Guardiano, *Evidence for syntax as a signal of historical relatedness*, Lingua 119 (2009) 1679–1706.

• Determiner Phrase Module:

- syntactic parameters dealing with person, number, gender (1–6)
- parameters of definiteness (7–16)
- parameters of countability (17–24)
- genitive structure (25–31)
- adjectival and relative modification (32–14)
- position and movement of the head noun (42–50)
- demonstratives and other determiners (51–50 and 6–63)
- possessive pronouns (56–59)

Simple Example:

• group of three languages $\mathcal{F} = \{\ell_1, \ell_2, \ell_3\}$: Italian, Spanish, French using first group of 6 parameters

• code $C = C(\mathcal{F})$

ℓ_1	1	1	1	0	1	1
ℓ_2	1	1	1	1	1	1
ℓ_3	1	1	1	0	1	0

• code parameters: $(R = \log_2(3)/6 = 0.2642, \delta = 1/6)$

• code parameters satisfy $R < 1 - H_2(\delta)$: below the Gilbert–Varshamov curve

Spoiling operations in this example:

• first spoiling operation:
first two parameters same value 1, so
$C = C' \star_1 f_1 = (C'' \star_2 f_2) \star_1 f_1$ with f_1 and f_2 constant equal to 1 and $C'' \subset \mathbb{F}_2^4$ without first two letters

• second spoiling operation:
conversely, $C'' = C' \star_2$ and $C' = C \star_1$

• third spoiling operation:
$C(0,4) = \{\ell_1, \ell_3\}$ and $C(1,6) = \{\ell_2, \ell_3\}$

What if languages are not in the same historical family?

Example: $\mathcal{F} = \{L_1, L_2, L_3\}$: Arabic, Wolof, Basque

• excluding parameters that are not set, or are entailed by other parameters, for these languages: left with 25 parameters from original list (number 1–5, 7, 10, 20–21, 25, 27–29, 31–32, 34, 37, 42, 50–53, 55–57)

• code $C = C(\mathcal{F})$

L_1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	1	1	1	1	1	0	1	0	0	0	0
L_2	1	1	1	0	0	1	1	0	1	0	1	0	0	1	0	1	1	0	0	1	1	1	1	1	1
L_3	1	1	0	1	0	0	1	0	0	0	1	1	1	0	1	1	0	1	1	1	1	1	1	0	0

• code parameters: $\delta = 0.52$ and $R > 0$ violates Plotkin bound
$\Rightarrow$ isolated code above the asymptotic bound

Asymptotic bound and language relatedness

• For binary syntactic parameters: a code $C = C(\mathcal{F})$ violates the Plotkin bound if any pair $L_i \neq L_j$ of languages in $\mathcal{F}$ has $\delta_H(L_i, L_j) \geq 1/2$

• L_i and L_j differ in at least half of the parameters: it would not happen in a group of historically related languages

• but what about codes above the asymptotic bound that do not violate the Plotkin bound?

• Expect: $C = C(\mathcal{F})$ above the asymptotic bound
$\Rightarrow \mathcal{F}$ not a historical language family
(quantitative test of historical relatedness)

Why the asymptotic bound?

• Why look at position with respect to asymptotic bound as a test of historical relatedness? because it is the only true "bound" in the space of code parameters across which behavior truly changes

• codes below the asymptotic bound are *easily deformable* (as long as number of syntactic parameters is large)

• if think of language evolution as a process of parameter change, expect languages that have evolved in the same family to determine codes in this zone of the space of code parameters

• codes $C = C(\mathcal{F})$ above the asymptotic bound should be a clear sign that list of languages in $\mathcal{F}$ do *not* belong to same historical family

• though there can be codes $C = C(\mathcal{F})$ below the asymptotic bound that also don't come from historically related languages: converse implication does not hold

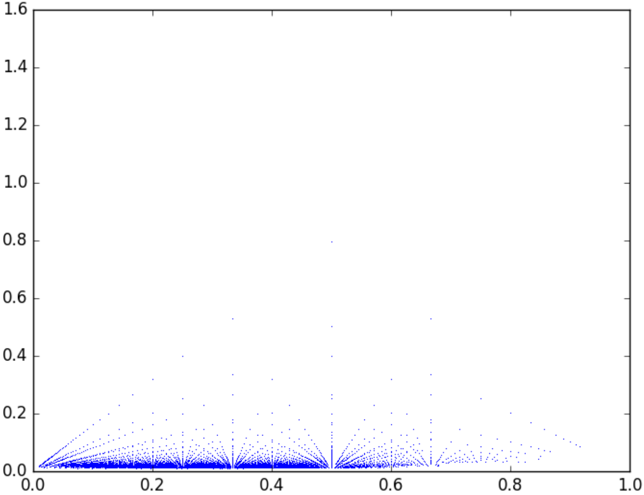
Code parameters of language sets

- Kevin Shu and Matilde Marcolli, *Syntactic structures and code parameters*, Mathematics in Computer Science 11 (2017) no. 1, 79–90.

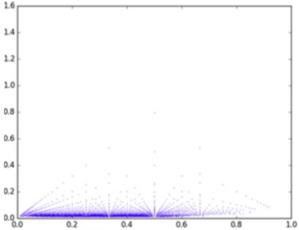
• take all sets of two and three languages in the SSWL database and set of parameters completely mapped for languages in the set

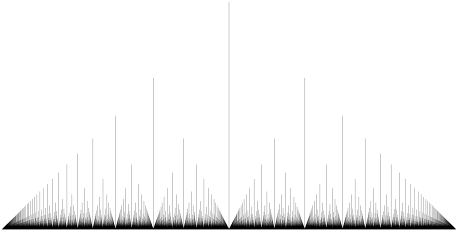
• for each pair/triple compute the code parameters of the resulting code and plot where they lie in the space of code parameters

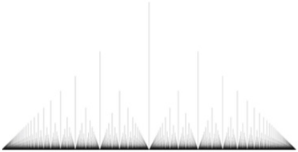
• distribution of code parameters for small sets of languages (pairs or triples) and SSWL data



- in lower region of code parameter space a superposition of two Thomae functions ($f(x) = 1/q$ for $x = p/q$ coprime, zero on irrationals)



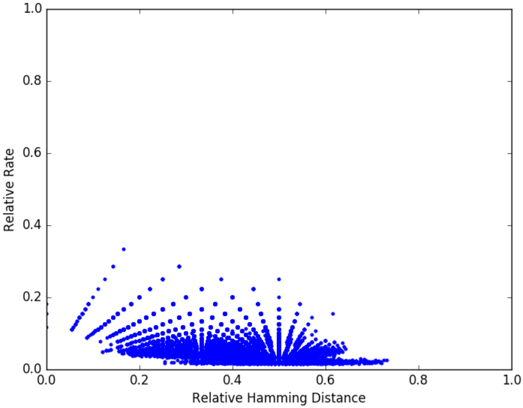




and behaves like the case of random codes with fixed $k = \log_2(N)$

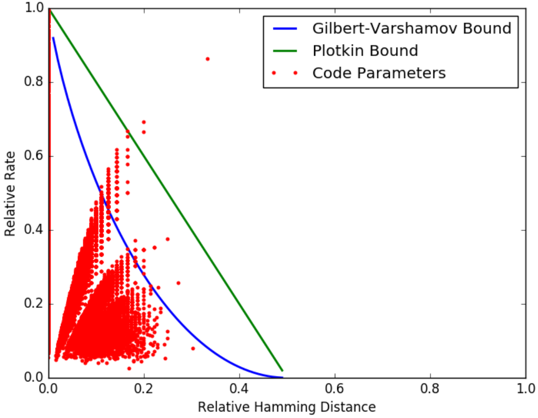
$$(\delta = \frac{d}{n}, R = k \cdot \frac{1}{n})$$

• randomly chosen sets of two or three languages tend to populate the lower region of the Thomae function graph



uniformly random sets of three languages

- more interesting what happens in the upper regions of the code parameter space
- take larger sets of randomly selected languages and syntactic parameters in the SSWL database



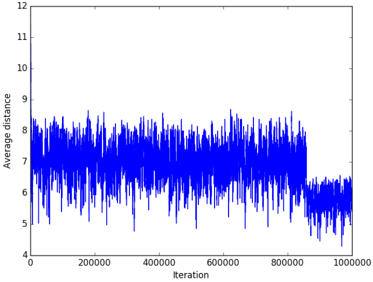
codes better than algebro-geometric above GV, asymptotic, and Plotkin

Space of Code Parameters and dynamics of syntactic parameters

• Spin Glass Model dynamics for a set of languages $\mathcal{L}$ induces dynamics on codes $\mathcal{C}(\mathcal{L})$ and on code parameters (R, δ)

- no entailment (independent parameters): fixed R and δ flows towards zero (spoiling code)
- entailment: dynamics can improve code making δ larger (R fixed)

• for large number of parameters see dynamics more easily on code parameter than with average magnetization of spin glass model

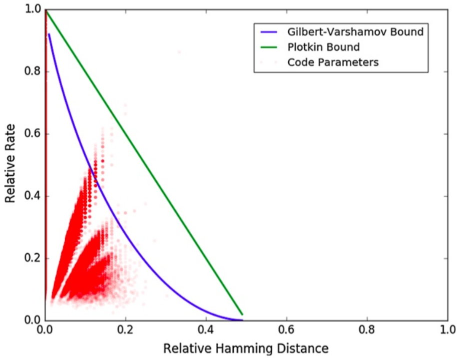


Remarks

• construction of binary codes above asymptotic bound through linguistics

• what are the best codes obtained this way? explicit examples with languages that are phylogenetically very distant

• these points are rare compared to typical: find explicitly which languages are involved



Syntactic Parameters in Kanerva Networks

• J.J. Park, R. Boettcher, A. Zhao, A. Mun, K. Yuh, V. Kumar, M. Marcolli, *Prevalence and recoverability of syntactic parameters in sparse distributed memories*, in "Geometric Science of Information. Third International Conference GSI 2017", pp. 265–272, Lecture Notes in Computer Science, Vol.10589, Springer 2017.

• Select a subset of SSWL parameters with properties:

- Completely mapped for a large number of languages in the database
- Known to have relations, though not of a simple explicit entailment form

• Detect which among these parameters are more or less recoverable from the other ones by testing recoverability in a sparse distributed memory

Preliminary considerations: Frequency of Expression

• different syntactic parameters have very different frequency of expression among world languages

• Example: Word Order: SOV, SVO, VSO, VOS, OVS, OSV

Word Orders	Percentage		
SOV	41.03%	Subject-initial	Specifier-Head
SVO	35.44%	Subject-initial	Specifier-Head
VSO	6.90%	Subject-medial	Head-Specifier
VOS	1.82%	Subject-final	Head-Specifier
OVS	0.79%	Subject-final	Head-Specifier
OSV	0.29%	Subject-medial	Specifier-Head

Very unevenly distributed across world languages

• this creates overall effect (using data that record expression of parameters among world languages): needs to be normalized when searching for abstract syntactic relations among parameters

Parameters and frequencies (as classified in SSWL)

01 Subject-Verb (0.64957267)
02 Verb-Subject (0.31623933)
03 Verb-Object (0.61538464)
04 Object-Verb (0.32478634)
05 Subject-Verb-Object (0.56837606)
06 Subject-Object-Verb (0.30769232)
07 Verb-Subject-Object (0.1923077)
08 Verb-Object-Subject (0.15811966)
09 Object-Subject-Verb (0.12393162)
10 Object-Verb-Subject (0.10683761)
11 Adposition-Noun-Phrase (0.58974361)
12 Noun-Phrase-Adposition (0.2905983)
13 Adjective-Noun (0.41025642)
14 Noun-Adjective (0.52564102)
15 Numeral-Noun (0.48290598)
16 Noun-Numeral (0.38034189)
17 Demonstrative-Noun (0.47435898)
18 Noun-Demonstrative (0.38461539)
19 Possessor-Noun (0.38034189)
20 Noun-Possessor (0.49145299)
A01 Attributive-Adjective-Agreement (0.46581197)

Kanerva networks (sparse distributed memories)

- P. Kanerva, *Sparse Distributed Memory*, MIT Press, 1988.
- field $\mathbb{F}_2 = \{0, 1\}$, vector space $\mathbb{F}_2^N$ large N
- uniform random sample of 2^k hard locations with $2^k << 2^N$
- median Hamming distance between hard locations
- Hamming spheres of radius slightly larger than median value (access sphere)
- *writing to network*: storing datum $X \in \mathbb{F}_2^N$, each hard location in access sphere of X gets i-th coordinate (initialized at zero) incremented depending on i-th entry ot X
- *reading at a location*: i-th entry determined by majority rule of i-th entries of all stored data in hard locations within access sphere

Kanerva networks are good at reconstructing corrupted data

Memory items in SDM: most items unrelated but most pairs linked by few intermediaries

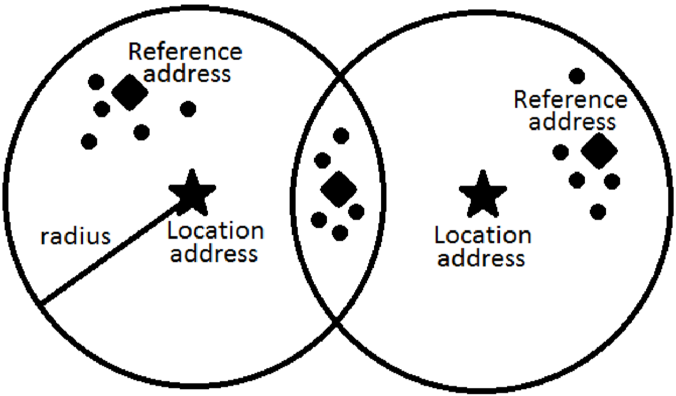


illustration from: Ján Kvak, *Creating and Recognizing Visual Words Using Sparse Distributed Memory* proposed as a realistic computational model of how information is stored and retrieved in human memory

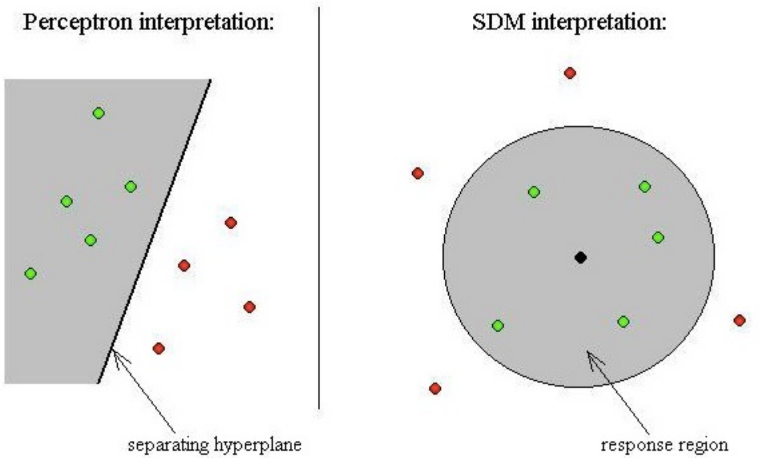
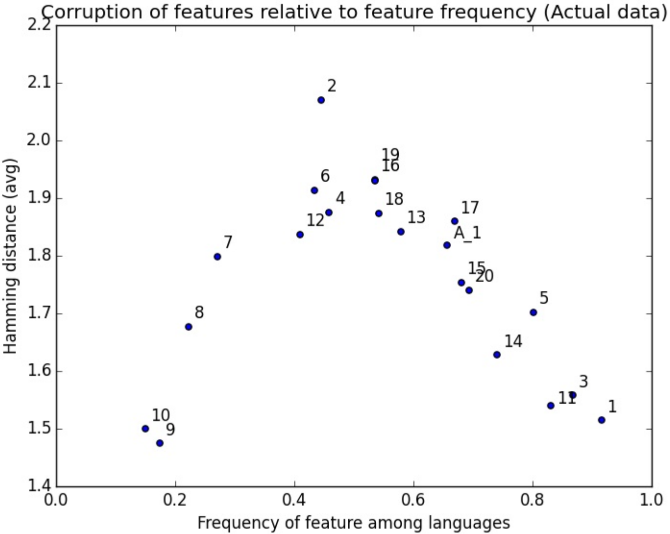


illustration from Jim Marshall's lecture notes on SDM

Procedure

- Kanerva Network with Boolean space $\mathbb{F}_2^{21}$
- 166 data points (fully mapped SSWL languages)
- Kanerva network with access sphere of $n/4$, with n median Hamming distance between points
- optimal: larger n excessive number of hard locations being in the sphere, computationally intractable
- correct data written to the Kanerva network
- known language bit-string, with a single corrupted bit, used as read location
- result of the read compared to original bit-string to test bit recovery
- average Hamming distance resulting from corruption of a given bit (a particular syntactic parameter) computed across all languages

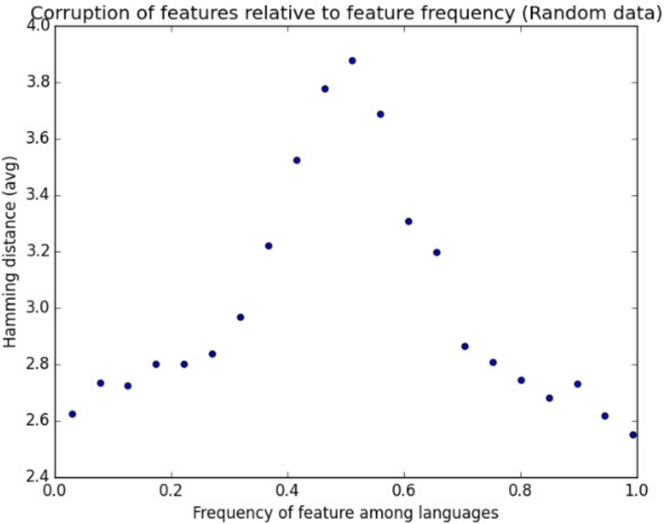
Recoverability in Kanerva Networks



need to identify effects due to syntax from overall frequency effect

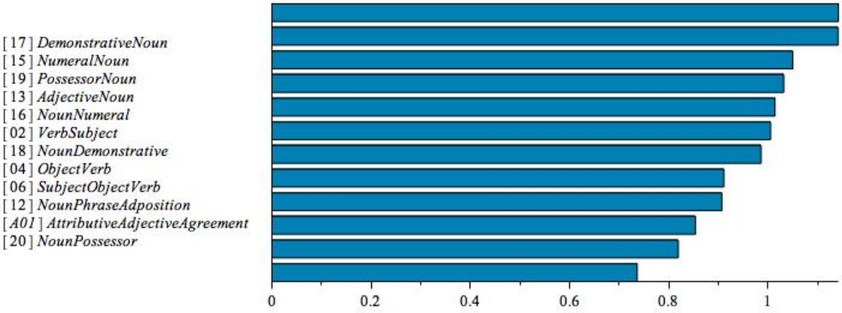
Normalize for frequency effect

- the recoverability data obtained combine two effects
  - an overall effect depending on the frequency of expression
  - a finer effect due to actual syntactic relations
- Procedure to separate overall frequency effect:
  - for each syntactic parameter subset of languages of fixed size chosen randomly with property that half of the languages have that parameter expressed
  - ignore those parameters with too few languages for which this can be done
  - use a fixed size of 95 languages
  - data of these languages written to Kanerva network and recoverability of corrupted individual parameters tested again
  - test run again with random data generated with an approximately similar distribution of bits



Overall effect related to relative prevalence of a parameter

More refined effect after normalizing for prelavence (extracting effect of syntactic dependencies)



Additional Remarks

- Overall effect relating recoverability in a Kanerva Network to prevalence of a certain parameter among languages (depends only on frequencies: see in random data with assigned frequencies)

- Additional effects (that deviate from random case) which detect possible dependencies among syntactic parameters: increased recoverability beyond what effect based on frequency

- Possible neuroscience implications? Kanerva Networks as models of human memory (parameter prevalence linked to neuroscience models)

- More refined effects if divided by language families?

Heat Kernel dimensional reduction

• Andrew Ortegaray, Robert C. Berwick, Matilde Marcolli, *Heat Kernel Analysis of Syntactic Structures*, arXiv:1803.09832

• Geometric methods of dimensional reduction: *Belkin–Niyogi heat kernel method*

• M. Belkin, P. Niyogi, *Laplacian eigenmaps for dimensionality reduction and data representation*, Neural Comput. 15 (6) (2003) 1373–1396.

• Question: low dimensional representations of data sampled from a probability distribution on a manifold

• Want more efficient methods than Principal Component Analysis

• Main Idea: build a graph with neighborhood information, use Laplacian of graph, obtain low dimensional representation that maintains the local neighborhood information using eigenfunctions of the Laplacian

Main idea of Belkin–Niyogi heat kernel method

- k-dimensional compact smooth manifold $\mathcal{M}$ isometrically embedded in some $\mathbb{R}^N$
- data $\mathcal{S} = \{x_1, \ldots, x_n\}$ sampled from a uniform distribution in the induced measure on $\mathcal{M}$
- associated graph Laplacian $L = L^{t,n} = D^{t,n} - W^{t,n}$

$$L_{t,n} f(x) = f(x) \sum_j \exp(-\frac{\|x - x_j\|^2}{4t}) - \sum_j f(x_j) \exp(-\frac{\|x - x_j\|^2}{4t})$$

- diagonal $D_{i,i} = D^{t,n}_{i,i} = \sum_j W^{t,n}_{i,j}$

Main Result: for sampled data $\mathcal{S} = \{x_1, \dots, x_n\}$ from uniform distribution on $\mathcal{M}$ take $t_n = n^{-(k+2+\alpha)^{-1}}$ with $\alpha > 0$: for some $C > 0$

$$\lim_{n\to\infty} C \frac{(4\pi t_n)^{-\frac{k+2}{2}}}{n} L^{t_n,n} f(x) = \Delta_M f(x)$$

for $f \in \mathcal{C}^\infty(\mathcal{M})$ with Δ_M = Laplace-Beltrami operator on $\mathcal{M}$

$$\Delta_{\mathcal{M}} f = \frac{1}{\sqrt{\det(g)}} \sum_j \frac{\partial}{\partial x^j} (\sqrt{\det(g)} \sum_i g^{ij} \frac{\partial}{\partial x^i} f)$$

g^{ij} inverse of the metric tensor

Laplace–Beltrami operator and heat kernel

- on $\mathbb{R}^N$

$$\Delta f(x) = \sum_i \frac{\partial^2}{\partial x_i^2} f(x)$$

heat kernel equation

$$\frac{\partial}{\partial t} u(x,t) = \Delta u(x,t)$$

solutions with initial heat distribution $f(x)$

$$H^t f(x) = \int_{\mathbb{R}^N} f(y) H^t(x,y) dy$$

convolution with heat kernel

$$H^t(x,y) = (4\pi t)^{-k/2} \exp(-\frac{\|x-y\|^2}{4t})$$

Heat kernel and approximating the Laplacian

- Laplacian and heat kernel:

$$-\Delta f(x) = \frac{\partial}{\partial t} H^t f(x)|_{t=0}$$

$$= \lim_{t\to 0} \frac{(4\pi t)^{-k/2}}{t} \int_{\mathbb{R}^N} e^{-\frac{\|x-y\|^2}{4t}} f(y) dy - \frac{(4\pi t)^{-k/2}}{t} f(x) \int_{\mathbb{R}^N} e^{-\frac{\|x-y\|^2}{4t}} dy$$

- approximation: (uniform sampling of y)

$$\frac{(4\pi t)^{-k/2}}{t\, n} (f(x) \sum_{i=1}^n e^{-\frac{\|y_i - x\|^2}{4t}} - \sum_{i=1}^n e^{-\frac{\|y_i - x\|^2}{4t}} f(y_i))$$

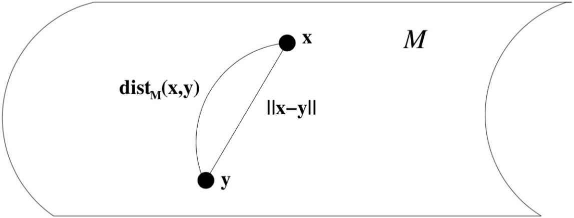
$$= C \, \frac{(4\pi t)^{-(k+2)/2}}{n} L^{t,n} f$$

- how to extend this idea from flat $\mathbb{R}^N$ to curved manifolds?

Laplacian approximation on manifolds

- geodesic distance and ambient Euclidean distance $\mathrm{dist}_{\mathcal{M}}(x, y) \geq \|x - y\|$
- exponential map $\exp_x : T_x\mathcal{M} \to \mathcal{M}$ takes lines through origin to geodesics
- on compact manifolds chord distance approximates geodesic distance

$$\mathrm{dist}_{\mathcal{M}}(x, y) = \|x - y\| + O(\|x - y\|)$$



Step 1: replace integral on $\mathcal{M}$ with integral on small open set $\mathcal{U}$ around a point $x \in \mathcal{M}$

• can do this because for $\mathcal{U} \subset \mathcal{M}$ open and $d^2 = \inf_{y \notin \mathcal{U}} \|x - y\|^2$

$$\left| \int_{\mathcal{U}} e^{-\frac{\|x-y\|^2}{4t}} f(y) d\mu_y - \int_{\mathcal{M}} e^{-\frac{\|x-y\|^2}{4t}} f(y) d\mu_y \right| \leq M \|f\|_\infty e^{-d^2/4t}$$

• then can use exponential map $v \mapsto \exp_x(v)$ to parameterize neighborhood $\mathcal{U}$ of $x \in \mathcal{M}$

• at point x where exp map centered

$$\Delta_{\mathcal{M}} f(x) = \Delta_{\mathbb{R}^k} \tilde{f}(0), \quad \tilde{f}(v) = f(\exp_x(v))$$

- S. Rosenberg, *The Laplacian on a Riemannian manifold*, Cambridge University Press, 1997.

The role of scalar curvature

- exp map locally invertible: $\mathcal{B} \subset \mathcal{U}$ with inverse, change coords

$$\int_{\mathcal{B}} e^{-\frac{\|x-y\|^2}{4t}} f(y) d\mu_y = \int_{\exp_x^{-1}(\mathcal{B})} e^{-\frac{\phi(v)}{4t}} \tilde{f}(v) \det(d\exp_x(v)) dv$$

with $\phi(v) = \|v\|^2 + O(\|v\|^4)$ (chord and geodesic dist)

- asymptotics of exp map

$$|\Delta_{\mathbb{R}^k} \det(d\exp_x(v))| = \frac{\kappa(x)}{3} + O(\|v\|)$$

κ scalar curvature

$$\Delta_{\mathbb{R}^k}(\tilde{f}\det(d\exp_x(v))(0) = \Delta_{\mathbb{R}^k}\tilde{f}(0) + k\frac{\kappa(x)}{3} f(x)$$

Cancellation of curvature terms

- then obtain

$$\frac{\partial}{\partial t}((4\pi t)^{-k/2}\int_{\mathcal{B}} e^{-\frac{\|x-y\|^2}{4t}} f(y) d\mu_y)|_{t=0} = \Delta_{\mathcal{M}} f(x) + \frac{k}{3}\kappa(x) f(x) + C f(x)$$

using previous and relation of $\Delta_{\mathcal{M}} f(x)$ and $\Delta_{\mathbb{R}^k} \tilde{f}(0)$

- then obtain

$$\lim_{t\to 0}(4\pi t)^{-k/2}(\int_{\mathcal{M}} e^{-\frac{\|x-y\|^2}{4t}} f(x) d\mu_y - \int_{\mathcal{M}} e^{-\frac{\|x-y\|^2}{4t}} f(y) d\mu_y) = \Delta_{\mathcal{M}} f(x)$$

- using sampling approximation for $\mathbb{R}^k$ case this gives

$$\lim_{n\to\infty}(4\pi t_n)^{-(k+2)/2} L^{t_n,n} f(x) = \frac{\Delta_{\mathcal{M}} f(x)}{\mathrm{Vol}(\mathcal{M})}$$

- this shows the graph Laplacian of a point cloud data set converges to the Laplace–Beltrami operator on the underlying manifold
- given map $f : \mathcal{M} \to \mathbb{R}$, points near x will map to points near $f(x)$ if gradient ∇f is sufficiently small
- minimizing square gradient reduces to finding eigenfunctions of the Laplace–Beltrami operator: Stokes theorem

$$\int_{\mathcal{M}} \|\nabla f\|^2 = \int_{\mathcal{M}} f \Delta_{\mathcal{M}} f$$

normalized local extrema are eigenfunctions

$$\lambda_n = \inf_{X_n} \frac{\int_{\mathcal{M}} \|\nabla f\|^2}{\int_{\mathcal{M}} f^2}$$

X_n complement of span of previous eigenfunctions

• Use to construct optimal mapping of data sets to low dimensional spaces via eigenfunctions of Laplacian

Algorithm

• setting: data points $x_1, \dots, x_k \in \mathcal{M} \subset \mathbb{R}^\ell$ on a manifold; find points $y_1, \dots, y_k$ in a low dimensional $\mathbb{R}^m$ ($m << \ell$) that *represent* the data points x_i

• Step 1 (a): adjacency graph (ϵ-neighborhood): an edge e_{ij} between x_i and x_j if $\|x_i - x_j\|_{\mathbb{R}^\ell} < \epsilon$

• Step 1 (b): adjacency graph (n nearest neighborhood): egde e_{ij} between x_i and x_j if x_i is among the n nearest neighbors of x_j or viceversa

• Step 2: weights on edges: heat kernel

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{t} \right)$$

if edge e_{ij} and $W_{ij} = 0$ otherwise; heat kernel parameter $t > 0$

• Step 3: Eigenfunctions for connected graph (or on each component)

$$L\psi = \lambda D\psi$$

diagonal matrix of weights $D_{ii} = \sum_j W_{ji}$; Laplacian $L = D - W$ with $W = (W_{ij})$; eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{k-1}$ and ψ_j eigenfuctions

$$\psi_i : \{1, \ldots, k\} \to \mathbb{R}$$

defined on set of vertices of graph

• Step 4: Mapping by Laplace eigenfunctions

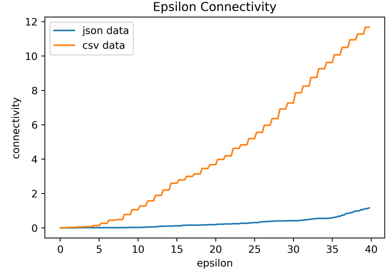
$$\mathbb{R}^\ell \supset \mathcal{M} \ni x_i \mapsto (\psi_1(i), \ldots, \psi_m(i)) \in \mathbb{R}^m$$

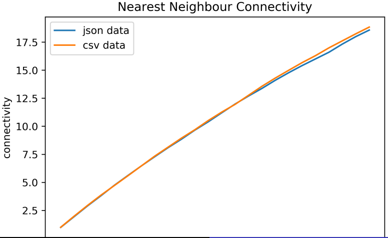
map by first m eigenfunctions

• Belkin–Niyogi: *optimality* of embedding by Laplace eigenfunctions

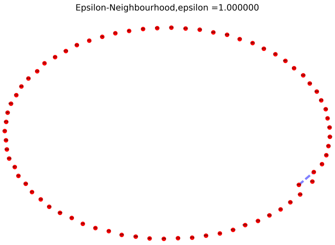
Heat Kernel analysis of Syntactic Parameters

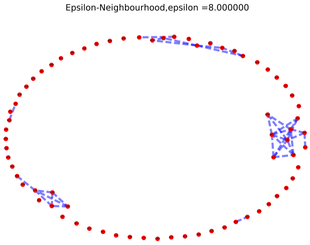
- Connectivity in ϵ-neighborhood and nearest-neighbor (difference between SSWL data (json) and Longobardi data (csv)



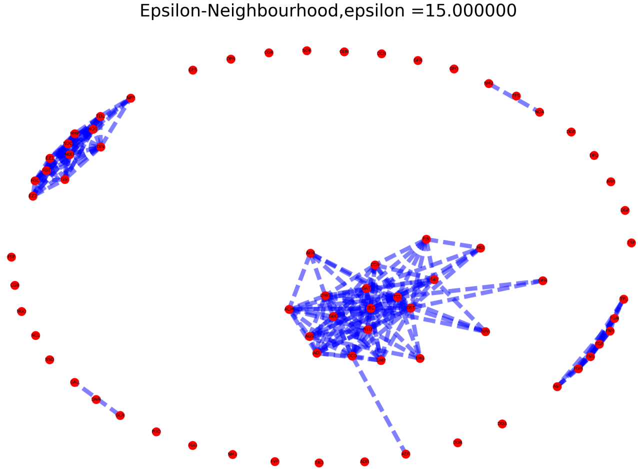


Graphs with ϵ-neighborhood Longobardi data

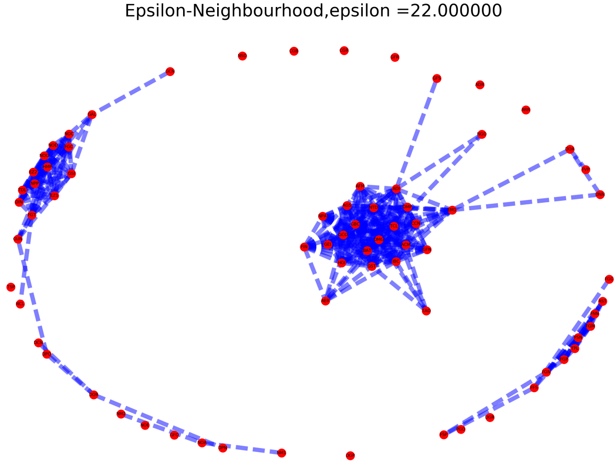




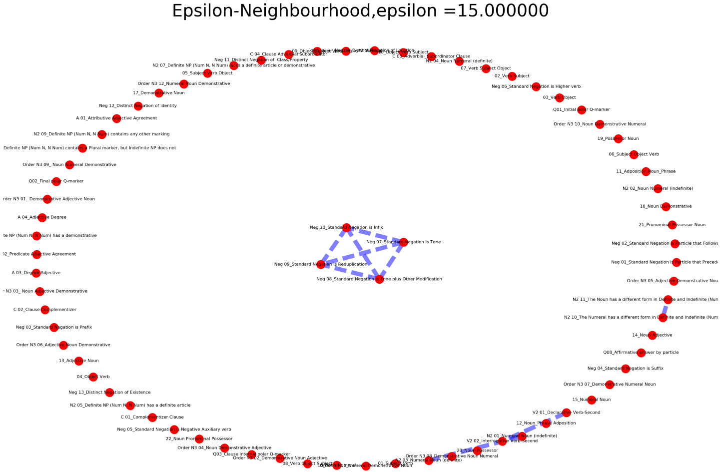
Graphs with ϵ-neighborhood Longobardi data



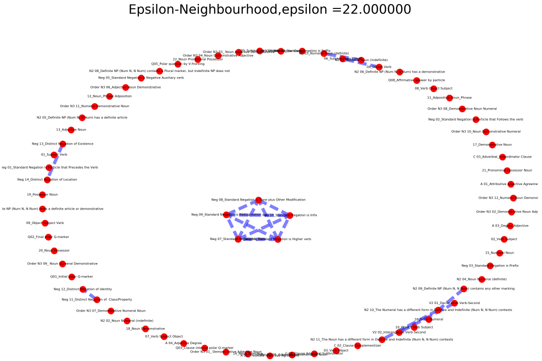
Graphs with ϵ-neighborhood Longobardi data



Graphs with ϵ-neighborhood SSWL data

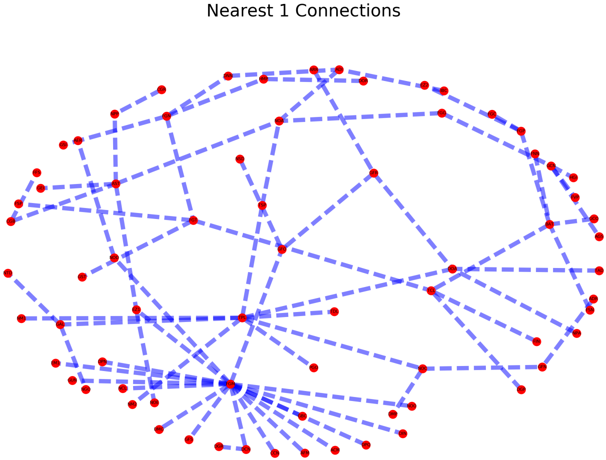


Graphs with ϵ-neighborhood SSWL data



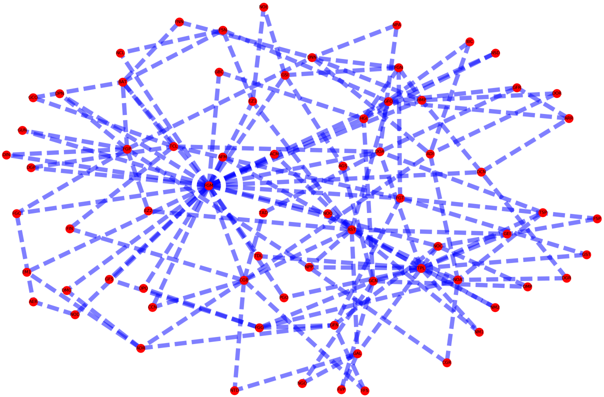
The ϵ-neighborhood construction is better suited to gain connectivity information in the Longobardi data: the SSWL data remain highly disconnected (only small local structures)

Graphs with *n*-neighborhood Longobardi data

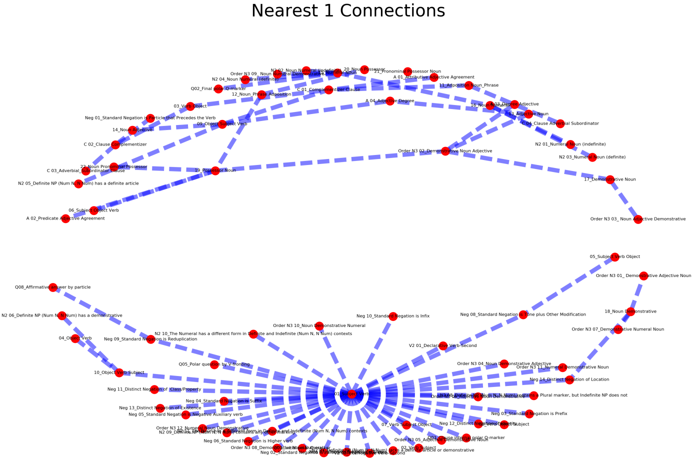


Graphs with *n*-neighborhood Longobardi data

Nearest 2 Connections

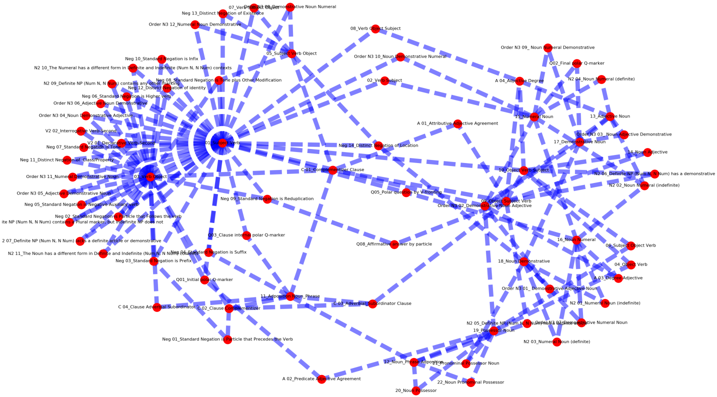


Graphs with *n*-neighborhood SSWL data



Graphs with *n*-neighborhood SSWL data

Nearest 2 Connections



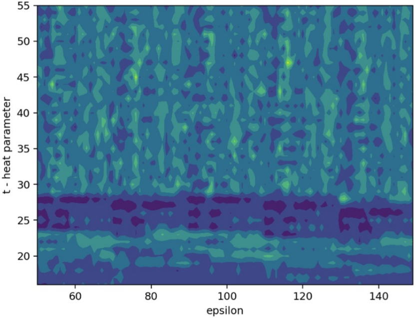
Regions of ϵ-t space

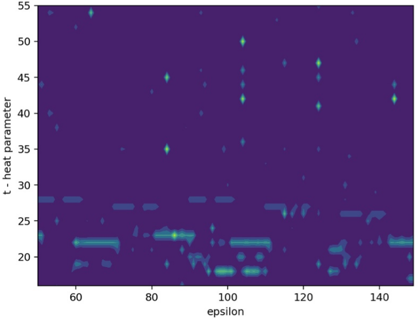
- Graphs depend on ϵ-neighborhood and on t-heat kernel variable

- explore ϵ-t space: determine regions where high variance in distribution of each parameter under the heat kernel mapping

- high variance in a parameter suggests it is highly independent (similar to PCA method)

- contour plots of variance; plots of number of outliers produced in set of coordinates for a given parameter


t - heat parameter
55
50
45
40
35
30
25
20
60
80
100
120
140
epsilon


t - heat parameter
55
50
45
40
35
30
25
20
60
80
100
120
140
epsilon

Further Questions

• an in depth linguistic analysis of the meaning of these clustering structures is still needed (ongoing work)

• comparison of the heat kernel technique with other dimensional reduction techniques (PCA etc.)

• more detailed discussion of different regions of the ϵ-t space in the heat kernel model (specific parameters with high independence measure)

• manifold $\mathcal{M}$ reconstruction? Belkin-Niyogi results