Geometrical morphology

John Goldsmith and Eric Rosen

February 24, 2017

Abstract

This paper explores a geometrical model for inflectional morphology in natural language. The basis of the underlying space are the values of the features represented in that morphology. The interface with the morphosyntax identifies a corner on a hypercube in this space, and the morpheme selected in order to realize that morphosyntactic selection consists of the morpheme whose vector representation is closest to that corner. In the case of a corner point realized by a set of inflectional morphemes, the set of morphemes that is chosen is the set whose vector sum is closest to the corner in question.

Preface

This is a working paper, describing work that the two of us have done over the course of 2016 and the beginning of 2017, developing in some detail an analysis of some aspects of inflectional morphology. In most respects this work continues the development of work in this area over the past fifty or more years, but in one or two important ways our approach is different. We believe that a *geometric* interpretation of inflectional morphology is possible, and that it simplifies our understanding of the formal aspects of the syntax-morphology interface. It simplifies our understanding in that renders unnecessary a large part of the derivational mechanics that have been part and parcel of most formalisms in this area. Readers for whom the mathematics below is unfamiliar are unlikely to agree with us that this is a simplification, but we will do our best to show that the mathematics and the geometry that we exploit is quite helpful in expressing our ideas about how languages work.

Here is a list of some central ideas, to get us started:

1. All linguists know that inflectional paradigms in natural language morphology are natural objects to think of as arrays in as few as 2 to as many as 10 or so dimensions. Each dimension corresponds to a linguistic feature, and features have a small number of possible values. The feature TENSE may have past/present as values, PERSON may have *1st/2nd/3rd* as values. But we have found that this is not the organization that is most useful for us.

2. It is sensible to view the dimensions of an inflectional paradigm as being built up out of a small set of privative features (features that take on the value *yes* or *no*, or 0/1). These privative features are the feature-values mentioned just above, like *1st person*, or *past tense*. Those features lead to feature-value space, of somewhat larger dimensionality than the arrays that linguists fashion for paradigms.

3. The set of morphosyntactic feature specifications that is specified by a grammar for a particular position in a sentence can be modeled as corners in feature-value space (i.e., points whose coordinates are all 0s and 1s). For example, a 3rd person singular present verb corresponds to a particular corner of a hypercube, whose coordinates are 1 for the dimensions *3rd-person*, *present*, and *singular*, but 0 for the four other dimensions. Not all corners in that space are meaningful, however: a point which takes the value 1 for both past tense and present tense will not be meaningful.

4. The morphemes that realize inflectional morphology can be analyzed as vectors in feature value space.

5. When we do so, we can say that the correct morpheme for any morphosyntactic feature specification (a *corner position* of a hypercube) is the morpheme that is *geometrically closest to it.*

6. When a word is composed of several morphemes, the proper choice of morphemes is the set of morphemes (each taken to be a vector) whose vector sum is geometrically closest to the given morphosyntactic feature specification.

7. Providing a learning mechanism by which the language-specific information about the grammar can be deduced or inferred from observed data is an essential aspect of a linguistic theory. Linguists' concerns about restricting the languages covered by linguistic theory are actually displaced (and misplaced) concerns about learnability.

8. Inflection patterns are sets of vectors in feature value space. Two related inflectional patterns are related to each by a rotation (that is, by a linear transformation that preserves inner product).

9. There is a (for us, surprising) similarity between the way this leads us to talk about morphology and the way in which quantum mechanics is described. And as a measurement in physics corresponds to a morpheme realization, we have no reason to view that as a probabilistic operation; instead, a prepared system is realized by the morpheme that is closest to it, not probabilistically by all morphemes onto which it could project.

Contents

1 Introduction

In this paper we explore a *geometric* way of understanding inflectional morphology.[1] The guiding idea is that many aspects of morphology can best be understood by thinking of features, morphemes, paradigms, and many other notions in morphology, through the lens of geometry. In a word: we can *visualize* morphology if properly conceived. At least we can if we can visualize spaces with quite a few dimensions to them. This geometrization of linguistic questions is similar in some regards to the advances achieved in phonology from better understanding the geometrical structure of phonological representations, and it is also motivated by the notion that grammatical explanation can be best achieved by showing that the correct forms generated by a grammar are those that maximize (or minimize) a (largely continuous) function defined over a high-dimensional space; they can therefore be understood in most cases as seeking a representation that maximizes or minimizes a particular function on representations.

One of the first characteristics of this approach is that we formalize the morphosyntactic feature specifications for a given inflected form not as a feature bundle, but as a point in a vector space. We will look at several different vector spaces: paradigm space, which is a slight variant on our traditional understanding of paradigms; feature-value space, which takes paradigm space apart into a larger space whose dimensions correspond to feature values; and morpheme space, in which the basis vectors are the morphemes (something that will become clear in what follows). Most of our work will take place in feature-value space.

In the cases we are interested in, the morphology serves as an interface between the morphosyntax and the morphological spelling-out. The morphosyntax specifies a corner point on a hypercube, and the morphology choses a vector (or, later, a set of vectors) that is as close to that corner point as possible.

1.1 Paradigm space

As we just observed, linguists often think about an inflectional paradigm as a multidimensional array, where each dimension takes on anywhere from 2 to 20 or so values (though it is uncommon for there to be more than 6 or so values for any given feature). One dimension might be number, with values singular and plural, and another might be person, with values 1st, 2nd and 3rd. The paradigm of the weak finite verb of English can be presented as in (1) (the presence of morpheme boundaries is not important at this point).

(1)

PERSON	*singular*	*plural*
	past	
1st	jump+ed	jump+ed
2nd	jump+ed	jump+ed
3rd	jump+ed	jump+ed

PERSON	*singular*	*plural*
	present	
1st	jump	jump
2nd	jump	jump
3rd	jump+s	jump

As a multidimensional array, each box or location in the array can be uniquely identified by a sequence of specifications in each dimension.[2] The box which contains *jump+s* can be identified as the position (Present, 3rd, singular). More generally, each position can be labeled as in (2).[3]

[1] This paper develops a proposal lightly sketched in Goldsmith (1994, 95-113). Stump (2016) is a good presentation of our starting point; we find his arguments clarifying the role of paradigms in morphology convincing, and if there is a current framework into which we imagine our work to be integrated, it is a framework such as the one that Stump develops there. We are grateful to discussions that have led to this work with a number of colleagues, including Stephen Fitz, Jackson Lee, Matt Goldrick, Doug Pulleyblank, Gunnar Hansson, Fred Chong, Risi Kondor, Brandon Rhodes, and Karlos Arregi.

[2] Stump (2016, 10-11) discusses some of the pros and cons of viewing a paradigm as a set of such boxes (or *cells*, as he calls them). Our view is that these cells are epiphenomena, even though the notion of a paradigm is not. That claim would probably be incoherent in the perspective he lays out; we explain our reasons over the course of this paper. Stump uses the phrase *inflectional category* where we use the more traditional term *feature*, and our *feature value* corresponds to his *morphosyntactic property*. We use what seems to us to be traditional terminology.

[3] Our use of the symbol ∅ should not be taken as a theoretical commitment on our part to the existence of null morphemes. We assume for the moment that such morphemes exist, and will return to the question below.

(2)

PERSON	*singular*	*plural*
PAST		
1st	(*past, 1st, singular*)	(*past, 1st, plural*)
2nd	(*past, 2nd, singular*)	(*past, 2nd, plural*)
3rd	(*past, 3rd, singular*)	(*past, 3rd, plural*)
PRESENT		
1st	(*present, 1st, singular*)	(*present, 1st, plural*)
2nd	(*present, 2nd, singular*)	(*present, 2nd, plural*)
3rd	(*present, 3rd, singular*)	(*present, 3rd, plural*)

However, while this representation is familiar to linguists, we will often find it more useful to express this information in a list-like way, as in (3), where we list the positions in the paradigm by ordering the dimensions of the paradigm in a fixed, conventional way (here, TENSE/PERSON/NUMBER). This is a representation in paradigm space. It is not a vector space, and not a space at all in a geometrical sense; the dimensions here are features (which are functions) that take on only one of a discrete set of values, and those values are not ordered among each other. (Thus despite the fact that we typically order 1st, 2nd, 3rd person in that order, the order is not reflected in the structure of the model.)

(3)

Paradigm space feature combination	verb	suffix
(past, 1st, singular)	jump+ed	ed
(past, 2nd, plural)	jump+ed	ed
(past, 3rd , singular)	jump+ed	ed
(past, 1st, plural)	jump+ed	ed
(past, 2nd, singular)	jump+ed	ed
(past, 3rd, plural)	jump+ed	ed
(present, 1st, singular)	jump	$\emptyset$
(present, 2nd, plural)	jump	$\emptyset$
(present, 3rd, singular)	jump+s	s
(present, 1st, plural)	jump	$\emptyset$
(present, 2nd, singular)	jump	$\emptyset$
(present, 3rd, plural)	jump	$\emptyset$

The principal role for paradigm space is as the interface between (morpho-)syntax and morphology. The syntax is concerned with specifying each syntactic position as one of the positions in paradigm space.

We can now construct a matrix which spells out the functions of each affix in the paradigm; we call this the *Total Paradigm Matrix*, or TPM. For the English weak verb, this is as in (4). The columns of TPM describe the positions in the paradigm in which each morpheme occurs, a set of positions in paradigm space; we indicate that vector with a hat over the morpheme, as in the following table.

(4) TOTAL PARADIGM MATRIX

	$\hat{\emptyset}$	$\hat{s}$	$\hat{ed}$
(past, 1st, singular)	0	0	1
(past, 2nd, plural)	0	0	1
(past, 3rd , singular)	0	0	1
(past, 1st, plural)	0	0	1
(past, 2nd, singular)	0	0	1
(past, 3rd, plural)	0	0	1
(present, 1st, singular)	1	0	0
(present, 2nd, plural)	1	0	0
(present, 3rd, singular)	0	1	0
(present, 1st, plural)	1	0	0
(present, 2nd, singular)	1	0	0
(present, 3rd, plural)	1	0	0

1.2 Feature-value space

Feature-value space has a different structure from paradigm space: it is the space in which most of our work in geometrical morphology takes place. In feature-value space, each dimension corresponds to a value taken on by one of the features in the paradigm, and for the present, we will consider only the values 0 and 1 that may be assigned to a coordinate. In the case of the English verb system, the example we will return to often initially, the feature-values are *past, present, 1st person, 2nd person, 3rd person, singular*, and *plural*. Thus in this example, feature-value space has 7 dimensions. A vector in that space is represented by a vector with 7 coordinates.

The dimensions are ordered arbitrarily; we adopt the convention that the feature values are ordered: (*Past, Present, 1st person, 2nd person, 3rd person, sg., pl.*). In (5), we give the feature-value space representation for each position in the verbal paradigm of English:

(5)

	Feature value space	
PERSON	*singular*	*plural*
PAST		
1st	(1,0,1,0,0,1,0)	(1,0,1,0,0,0,1)
2nd	(1,0,0,1,0,1,0)	(1,0,0,1,0,0,1)
3rd	(1,0,0,0,1,1,0)	(1,0,0,0,1,0,1)
PRESENT		
1st	(0,1,1,0,0,1,0)	(0,1,1,0,0,0,1)
2nd	(0,1,0,1,0,1,0)	(0,1,0,1,0,0,1)
3rd	(0,1,0,0,1,1,0)	(0,1,0,0,1,0,1)

Those 12 corner points on a 7-dimensional cube could be thought of as messages coming from the syntax, or morphosyntax, if one likes a dynamic metaphor. In any event, this "message" is limited to these corner points on the hypercube: only values of 0 and 1 for each dimension.

In (6) we present a binary matrix that shows the relationship between the positions in the paradigm and the coordinates in feature value space, and we call it Φ, as we indicate there. Each row is a possible "message" to be expressed; each column corresponds to a particular feature value.

(6)

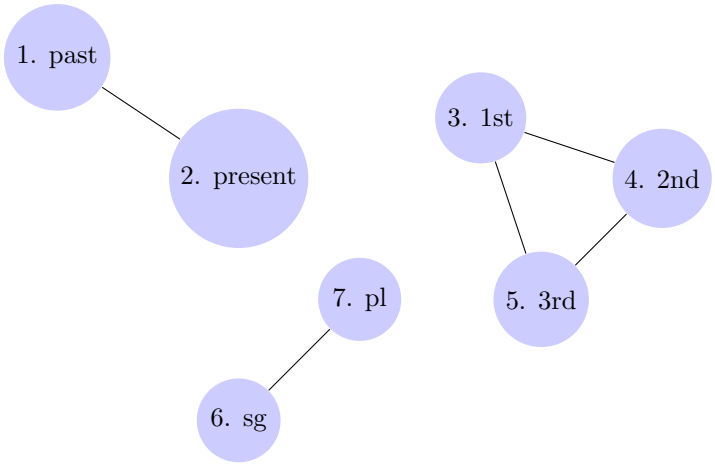
	past	*present*	*1st*	*2nd*	*3rd*	*singular*	*plural*
Past,1st, sg	1	0	1	0	0	1	0
Past,2nd, sg	1	0	0	1	0	1	0
Past,3rd, sg	1	0	0	0	1	1	0
Past,1st, pl	1	0	1	0	0	0	1
Past,2nd, pl	1	0	0	1	0	0	1
Past,3rd, pl	1	0	0	0	1	0	1
Present,1st, sg	0	1	1	0	0	1	0
Present,2nd, sg	0	1	0	1	0	1	0
Present,3rd, sg	0	1	0	0	1	1	0
Present,1st, pl	0	1	1	0	0	0	1
Present,2nd, pl	0	1	0	1	0	0	1
Present,3rd, pl	0	1	0	0	1	0	1

$= \Phi$

The columns (which express the feature values for each morpheme) divide up into sets which have the property that any two columns in the same set are orthogonal to each other, that is, that the inner product of any two is zero. In the case here, these sets are the first two columns, the next three columns, and the last two columns. For example, the inner product of the column vector for past and that for present is zero, since for each row, one vector or the other has the value 0. In addition, the sum of all of the vectors in each subset is the identity vector 1^7. These subsets correspond to the linguist's conception of a *feature*, and there are 3 here: TENSE, PERSON, and NUMBER.[4]

Another way to think about Φ involves the eigenmaps on the graph that represents the relationship between features and feature values. A graph whose nodes are feature values, and which contains edges between any two nodes whose feature values are values of the same feature, looks like this:

[4]See Goldsmith and Rosen for discussion of how such featural systems are learned, in a study of gender and case in German.



The eigenmaps on this graph are (1,1,0,0,0,0,0), (0,0,1,1,1,0,0), and (0,0,0,0,0,1,1), each corresponding to a linguistic features (tense, person and number, respectively).[5]

The two objects for the reader to remember now are the Total Paradigm Matrix (TPM), and Φ. The TPM expresses information about the paradigm, while Φ expresses the relationship between paradigm space and feature-value space.

1.3 Corners and the unit sphere in feature-value space

1.3.1 Corners in feature-value space

In feature-value space, there are certain points that are special, because they represent positions in the paradigm; they are the points whose coordinates appear as rows in the matrix Φ. In the case of the English verb, these *corners* are the points that have exactly one 1 in the first two coordinates, exactly one 1 in the next three coordinates, and exactly one 1 in the final two coordinates (all other coordinates are 0).

In addition, each inflectional morpheme is represented by a vector in feature-value space. As we will see, the three verbal suffixes *ed*, *s*, and $\emptyset$ each correspond to vectors there. The problem of morpheme selection thus can be restated as: for each corner, find the appropriate morpheme. Our hypothesis is that

> For each corner, the correct morpheme is the one that is closest to that corner.

1.3.2 The unit sphere in feature value space

One of the central ideas of this approach is that the morphological characteristics of a morpheme are best understood in the feature-value space. Each morpheme is modeled by a vector from the origin to a position in the feature-value space. Furthermore, we take all morphemes to be associated with vectors of unit length, i.e., the length $|\mu|$ of a morpheme is equal to $\sqrt{\sum_i \mu_i^2}$; these vectors reach from the origin to a point on the surface of a hypersphere of unit radius.[6]

Thus (1,0,0,0,1,0) is not a possible vector for a morpheme, because its length is $\sqrt{2}$. A morpheme that represents the feature-value "singular" can be represented as (1,0,0,0,0,0) since its length is 1.0. A morpheme which is specified as *nominative singular* can be represented as $(\frac{\sqrt{2}}{2}, 0, \frac{\sqrt{2}}{2}, 0, 0)$. The crucial point is that *morphosyntactic positions* are corners, that is, points whose coordinates are 0s and 1s

[5] An eigenmap on a graph is a function f from nodes to reals which is mapped to a multiple of f by the laplacian of the graph. The laplacian of a graph is an operator on functions on graphs (i.e., on maps from nodes to reals) which can be thought of as a rigorous account of how an activation on a set of nodes would diffuse through the graph, if edge weight were understood as a measurement of channel width. If a graph has disconnected subgraphs, then for each disconnected subgraph there will be at least one eigenmap that are zero on all the other disconnected subgraphs.

[6] It is very often the case that when we consider vectors in a high-dimensional space, when we speak of how close two vectors are, we do not mean the distance from the tip of one to the tip of the other, but care rather about the angle between the vectors. In such cases, it is common to say that we *normalize* the vectors, which is simply to shrink them to length 1 while maintaining their direction. Bear in mind that when we speak of *normalization*, we assume a particular notion of length or dimensionality. When we are dealing with probabilities, normalization generally refers to finding a scaling factor so that the sum of the values (here, the coordinates) sum to 1.0. This employs what is called the L_1 norm. When we think geometrically, we employ the L_2 norm, which means that normalization divides each term by a factor so that the sum of the squares of those normalized terms sum to 1.0

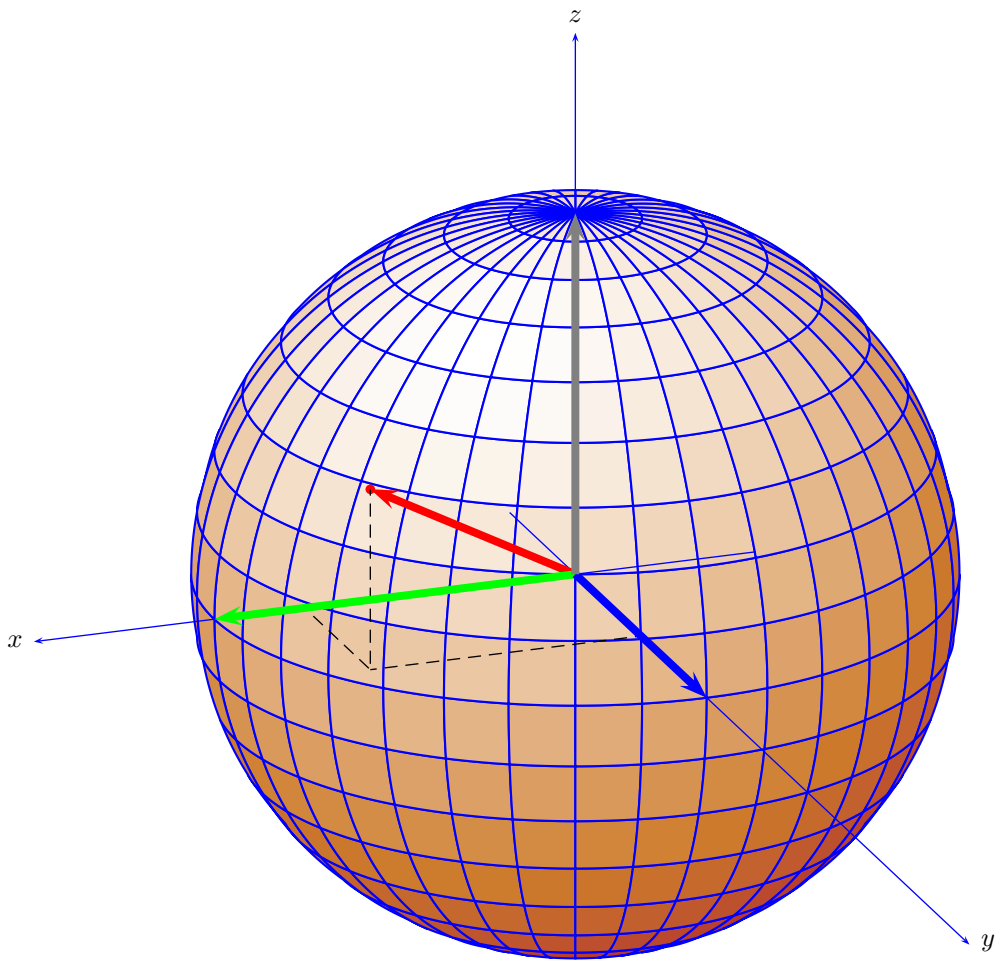


Figure 1: A sphere of radius 1, with four vectors of length 1.0

(subject to constraints), while *morphemes* are vectors that typically are *not* composed of only 0s and 1s: they are of unit length, however. Typically they take on positive values for those feature-values that they realize, for reasons that will emerge in our discussion below. The coordinates may take on negative values as well.[7]

The correct morpheme for each position is the morpheme which is *closest* to the position, in the geometric sense, in the feature-value space. We define closeness as the inner product of the two vectors (one for the position in the paradigm, one for the morpheme candidate).

These two vectors play different sorts of roles. The vectors representing morphemes are normalized to unit length, and are directly linked to observable characteristics of the word. The positions in a paradigm, that is to say the specification of the morphosyntactic features, is typically not of unit length, and is in a certain sense hidden from the linguist.

We will see below that it is not always a *single* vector that we are looking for, one which is as close as possible to the target corner; it will in general be a set of vectors that we seek, whose vector sum is as close as possible. When a word consists of several morphemes, the vector corresponding to that multimorphemic word is the (vector) sum of the vectors corresponding to each morpheme. We will explore this in section 3.

We frequently compute the inner product of two such vectors, and in order to emphasize the difference between the two, we will make use of a notation in which the observable morpheme is placed thusly: $\langle\mu|$ and the morphosyntactic specification is written thusly: $|M\rangle$ (this notation is known as Dirac's bracket notation). The inner product of the two is written $\langle\mu|\,|M\rangle$. The central formal idea that we develop in this paper employs the following choice procedure:[8]

$$\hat{\mu} = argmax_i \langle\mu_i|\,|M\rangle$$

where $\hat{\mu}$ is the morpheme selected by the grammar.

We will hold in abeyance for just a moment how the vector that is associated with a given morpheme is determined; for the moment, consider the following matrix, whose column vectors specify the vectors for each affix in the English weak verb conjugation (7). We refer to this matrix as $\mathcal{B}$, and the reader can easily see that the column vectors are of unit length (the sum of the squares of their coordinates sum to 1.0).

(7)
$$\begin{array}{l} \\ \textit{past} \\ \textit{present} \\ \textit{1st} \\ \textit{2nd} \\ \textit{3rd} \\ \textit{sg} \\ \textit{pl} \end{array} \begin{array}{c} \begin{array}{ccc} \emptyset & -s & -ed \end{array} \\ \begin{pmatrix} 0 & 0 & \frac{6}{\sqrt{66}} \\ \frac{5}{\sqrt{47}} & \frac{1}{\sqrt{3}} & 0 \\ \frac{2}{\sqrt{47}} & 0 & \frac{2}{\sqrt{66}} \\ \frac{2}{\sqrt{47}} & 0 & \frac{2}{\sqrt{66}} \\ \frac{1}{\sqrt{47}} & \frac{1}{\sqrt{3}} & \frac{2}{\sqrt{66}} \\ \frac{2}{\sqrt{47}} & \frac{1}{\sqrt{3}} & \frac{3}{\sqrt{66}} \\ \frac{3}{\sqrt{47}} & 0 & \frac{3}{\sqrt{66}} \end{pmatrix} \end{array} = \mathcal{B}$$

which is approximately:

[7] As we were writing this, we by good fortune came across what Scott Aaronson wrote in *Quantum Computing Since Democritus*: "Now what happens if you try to come up with a theory that's *like* probability theory, but based on the 2-norm instead of on the 1-norm? I'm going to try to convince you that quantum mechanics is what inevitably results." (Aaronson 2013, 112)

[8] The bracket notation is intended to remind the reader of treating this as a collapse of the wave function, where M represents the state of a system, and the morphemes essentially *are* observation operations, and rather than assigning probability based on the inner product, the choice is deterministic; the morpheme (observation) with the largest inner product (hence, smallest angle) is predicted. Seaking the largest inner product, which we speak of *maximizing*, is often equivalent to choosing the vector with the smallest angle compared to some particular fixed vector, and so sometimes we will speak of minimizing an angle, which in the cases we consider is the same thing as maximizing the inner product.

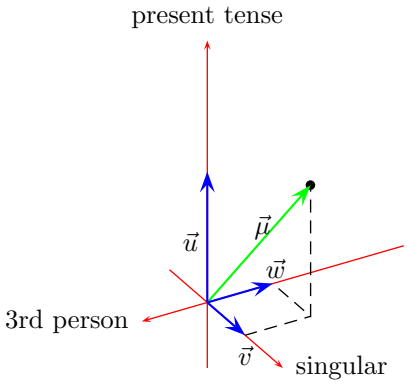


Figure 2: A vector sum

$$(8) \quad \mathcal{B} \approx \begin{bmatrix} 0 & 0 & .738 \\ .739 & .577 & 0 \\ .292 & 0.0 & 0.246 \\ .292 & 0 & 0.246 \\ .146 & .577 & .246 \\ .292 & .577 & .369 \\ .438 & 0 & .369 \end{bmatrix}$$

1.4 From $\mathcal{B}$ to morpheme selection

Let's see how this plays out for selection of the suffix in the present tense. We calculate the product $\Phi\times\ \mathcal{B}$, which in effect gives us the inner product of each morpheme's vector (in (7)) with its position vector (in (6)); we show only two decimal places because with even three, the table becomes visually unreadable.

	∅	-s	-ed
past 1st sg	(1,0, 1,0,0, 1,0)(0,.73, .29,.29,.15, .29,.44)	(1,0, 1,0,0, 1,0)(0,.58, 0,0,.58, .58,0)	(1,0, 1,0,0, 1,0)(.74,0, .25,,25,.25, .37,.37)
past 2nd sg	(1,0, 0,1,0, 1,0)(0,.73, .29,.29,.15, .29,.44)	(1,0, 0,1,0, 1,0)(0,.58, 0,0,.58, .58,0)	(1,0, 0,1,0, 1,0)(.74,0, .25,,25,.25, .37,.37)
past 3rd sg	(1,0, 0,0,1, 1,0)(0,.73, .29,.29,.15, .29,.44)	(1,0, 0,0,1, 1,0)(0,.58, 0,0,.58, .58,0)	(1,0, 0,0,1, 1,0)(.74,0, .25,,25,.25, .37,.37)
past 1st pl	(1,0, 1,0,0, 0,1)(0,.73, .29,.29,.15, .29,.44)	(1,0, 1,0,0, 0,1)(0,.58, 0,0,.58, .58,0)	(1,0, 1,0,0, 0,1)(.74,0, .25,,25,.25, .37,.37)
past 2nd pl	(1,0, 0,1,0, 0,1)(0,.73, .29,.29,.15, .29,.44)	(1,0, 0,1,0, 0,1)(0,.58, 0,0,.58, .58,0)	(1,0, 0,1,0, 0,1)(.74,0, .25,,25,.25, .37,.37)
past 3rd pl	(1,0, 0,0,1, 0,1)(0,.73, .29,.29,.15, .29,.44)	(1,0, 0,0,1, 0,1)(0,.58, 0,0,.58, .58,0)	(1,0, 0,0,1, 0,1)(.74,0, .25,,25,.25, .37,.37)
present 1st sg	(0,1, 1,0,0, 1,0)(0,.73, .29,.29,.15, .29,.44)	(0,1, 1,0,0, 1,0)(0,.58, 0,0,.58, .58,0)	(0,1, 1,0,0, 1,0)(.74,0, .25,,25,.25, .37,.37)
present 2nd sg	(0,1, 0,1,0, 1,0)(0,.73, .29,.29,.15, .29,.44)	(0,1, 0,1,0, 1,0)(0,.58, 0,0,.58, .58,0)	(0,1, 0,1,0, 1,0)(.74,0, .25,,25,.25, .37,.37)
present 3rd sg	(0,1, 0,0,1, 1,0)(0,.73, .29,.29,.15, .29,.44)	(0,1, 0,0,1, 1,0)(0,.58, 0,0,.58, .58,0)	(0,1, 0,0,1, 1,0)(.74,0, .25,,25,.25, .37,.37)
present 1st pl	(0,1, 1,0,0, 0,1)(0,.73, .29,.29,.15, .29,.44)	(0,1, 1,0,0, 0,1)(0,.58, 0,0,.58, .58,0)	(0,1, 1,0,0, 0,1)(.74,0, .25,,25,.25, .37,.37)
present 2nd pl	(0,1, 0,1,0, 0,1)(0,.73, .29,.29,.15, .29,.44)	(0,1, 0,1,0, 0,1)(0,.58, 0,0,.58, .58,0)	(0,1, 0,1,0, 0,1)(.74,0, .25,,25,.25, .37,.37)
present 3rd pl	(0,1, 0,0,1, 0,1)(0,.73, .29,.29,.15, .29,.44)	(0,1, 0,0,1, 0,1)(0,.58, 0,0,.58, .58,0)	(0,1, 0,0,1, 0,1)(.74,0, .25,,25,.25, .37,.37)

Table 1: $\Phi\times\ \mathcal{B}$, Weak verb

which equals the following (where we have put in blue the largest value in each row, marking the winning suffix):

(9) $\Phi \times \mathcal{B}$=

suffix	∅	-s	-ed
1st sg past	0.584	0.577	1.353
2nd sg past	0.584	0.577	1.353
3rd sg past	0.438	1.154	1.353
1st pl past	0.730	0	1.353
2nd pl past	0.730	0	1.353
3rd pl past	0.584	0.577	1.353
1st sg pres	1.313	1.154	0.615
2nd sg pres	1.313	1.154	0.615
3rd sg pres	1.167	1.731	0.615
1st pl pres	1.459	0.577	0.615
2nd pl pres	1.459	0.577	0.615
3rd pl pres	1.313	1.154	0.615

We have already indicated that choosing the element with the highest value in each row will be a central part of our analysis, and so it is convenient to be able to refer to that element. We use a notation Max_{rows}(M,i,j) which we define as:

$$Max_{rows}(M,i,j) = \begin{cases} 1 & \text{if for all k} \neq \text{j, M(i,j)> M(i,k)} \\ 0 & \text{otherwise.} \end{cases}$$

and more generally, extending that function to the whole matrix, so that we may write $Max_{rows}(M)$:

$$Max_{rows}(M)(i,j) = Max_{rows}(M,i,j)$$

With this notation, our colored, winning entries in $\Phi\times\ \mathcal{B}$ take the value 1 in $Min_{row}(\Phi \times \mathcal{B})$. This is what we have already labeled the *Total Paradigm Matrix* (TPM). What we have just shown is that the TPM can be computed from Φ and $\mathcal{B}$; this is *not* a mathematical certainty (we will see cases where a value of $\mathcal{B}$ is maximal and makes an incorrect prediction); there is always an empirical test as to whether a particular set of values for the morphemes ($\mathcal{B}$) correctly leads to the TPM.[9]

(10) TPM: $Max_{rows}(\Phi \times \mathcal{B}) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

1.5 Smart initialization

One of the most important aspects of our approach is its connection to a theory of learning. It is our goal to develop a model of morphology which stands or falls on its ability to project, or induce, a satisfactory analysis from data. As before, we take it for granted that we are in possession of the morphosyntactic features of the system, and that the principal problem is how we assign a vector to each of the suffixes.

Let's return to the table in (6), which we repeat here as (11), the matrix Φ that has NumParaPos rows, and NumFeaVal columns. Each row in Φ corresponds to a position in the paradigm. A morpheme can be represented initially as a (column) vector with a 1 in each position which it represents in the paradigm. The whole paradigm is represented as a matrix, which we will call Φ. As we will see shortly, Φ is a linear transformation that maps a paradigm representation (see below) to an element in feature-value space.

[9] We use the term "empirical" here as it is often used in linguistics (though it would be hard to justify this usage outside linguistics) whereby a statement is empirical iff it could be shown to be wrong.

(11)

	past	*present*	*1st*	*2nd*	*3rd*	*singular*	*plural*
Past,1st, sg	1	0	1	0	0	1	0
Past,2nd, sg	1	0	0	1	0	1	0
Past,3rd, sg	1	0	0	0	1	1	0
Past,1st, pl	1	0	1	0	0	0	1
Past,2nd, pl	1	0	0	1	0	0	1
Past,3rd, pl	1	0	0	0	1	0	1
Present,1st, sg	0	1	1	0	0	1	0
Present,2nd, sg	0	1	0	1	0	1	0
Present,3rd, sg	0	1	0	0	1	1	0
Present,1st, pl	0	1	1	0	0	0	1
Present,2nd, pl	0	1	0	1	0	0	1
Present,3rd, pl	0	1	0	0	1	0	1

$= \Phi$

The three morphemes $\emptyset$, s, and ed can be represented as vectors in this way. Here we are representing them as the points in the paradigm that they represent, so there are 6 rows, one for each position in the paradigm. We call these *paradigm representations*, and mark them with a hat: $\hat{f}$.

(12) $\hat{\emptyset} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \hat{s} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \hat{ed} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

If we take the matrix product of Φ^t (Φ transpose) and each paradigm representation, we get a summary of each morpheme's expression of each feature-value. Here are the values that result. The vector that results exists in the feature-value space. For example, the first product shows that the null morpheme has a 0 value in the *past*, 5 in the *present*, 2 in *1st person*, 2 in *2nd person*, 1 in *3rd person*, 2 in the *singular*, and 3 in the *plural*. In short: Φ^t maps from *PS* (paradigm space) to *FV* (feature value space).

(13) $\langle\Phi|\,|\emptyset\rangle = \Phi^t\hat{\emptyset} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 2 \\ 2 \\ 1 \\ 2 \\ 3 \end{bmatrix}$

Similarly,

(14) $\langle\Phi|\,|s\rangle = \Phi^t\hat{s} = \begin{bmatrix} 1&1&1&1&1&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&1&1&1&1&1\\ 1&0&0&1&0&0&1&0&0&1&0&0\\ 0&1&0&0&1&0&0&1&0&0&1&0\\ 0&0&1&0&0&1&0&0&1&0&0&1\\ 1&1&1&0&0&0&1&1&1&0&0&0\\ 0&0&0&1&1&1&0&0&0&1&1&1 \end{bmatrix} \begin{bmatrix} 0\\0\\0\\0\\0\\0\\0\\0\\1\\0\\0\\0 \end{bmatrix} = \begin{bmatrix} 0\\1\\0\\0\\1\\1\\0 \end{bmatrix}$

s has a positive value in present, 3rd person, and singular.

(15) $\langle\Phi|\,|ed\rangle = \Phi^t\hat{ed} = \begin{bmatrix} 1&1&1&1&1&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&1&1&1&1&1\\ 1&0&0&1&0&0&1&0&0&1&0&0\\ 0&1&0&0&1&0&0&1&0&0&1&0\\ 0&0&1&0&0&1&0&0&1&0&0&1\\ 1&1&1&0&0&0&1&1&1&0&0&0\\ 0&0&0&1&1&1&0&0&0&1&1&1 \end{bmatrix} \begin{bmatrix} 1\\1\\1\\1\\1\\1\\1\\0\\0\\0\\0\\0 \end{bmatrix} = \begin{bmatrix} 6\\0\\2\\2\\2\\3\\3 \end{bmatrix}$

If matrix operations are not familiar to the reader, this operation corresponds to this: count the number of times each morphosyntactic features occurs with the realization of each suffix:

(16)

"Count array"	∅	-s	-ed
past	0	0	6
present	5	1	0
1st	2	0	2
2nd	2	0	2
3rd	1	1	2
singular	2	1	3
plural	3	0	3

Now we normalize each column vector (one for each morpheme), making it of unit length, and we arrive at the tables given just above in (7). Having done that, we can use the *bra* $\langle.|$ notation.

(17)

	$\langle\emptyset\|$	$\langle -s\|$	$\langle -ed\|$
past	0	0	$\frac{6}{\sqrt{66}}$
present	$\frac{5}{\sqrt{47}}$	$\frac{1}{\sqrt{3}}$	0
1st	$\frac{2}{\sqrt{47}}$	0	$\frac{2}{\sqrt{66}}$
2nd	$\frac{2}{\sqrt{47}}$	0	$\frac{2}{\sqrt{66}}$
3rd	$\frac{1}{\sqrt{47}}$	$\frac{1}{\sqrt{3}}$	$\frac{2}{\sqrt{66}}$
sg	$\frac{2}{\sqrt{47}}$	$\frac{1}{\sqrt{3}}$	$\frac{3}{\sqrt{66}}$
pl	$\frac{3}{\sqrt{47}}$	0	$\frac{3}{\sqrt{66}}$

$= \mathcal{B}$

These values represent hypothesized settings for the morphemes derived directly from one occurrence of each point in the paradigm. The same operation can be performed with even an incomplete set

of data from the paradigm. In either case, there is no guarantee that the morpheme-vectors correctly generate the appropriate forms for each point in the paradigm, which is why we refer to this process as *initialization*. We will explore shortly another learning process that we employ to deal with the cases in which smart initialization is not enough, a traditional learning rule called the Delta Rule.

Let us summarize what we have done. We have used a vector/matrix notation to describe part of a relatively simple inflectional paradigm, and we have begun by assuming that we know the correct morpheme for each position in the paradigm. On the basis of that knowledge, we infer a vectorial representation for the three morphemes. Each box in the traditional paradigm corresponds to a corner of a hypercube in feature value space, and the morpheme that is closest to that corner is the correct morpheme for that position in the paradigm.

1.6 Syncretism and blocking

One of the characteristics of the present model which we find the most striking is the naturalness of what has traditionally been called *syncretism*, which is the use of the same inflectional morpheme to represent more than one position (Stump's *cell*) in the inflectional paradigm. This phenomenon has a natural interpretation in the present context: a particular morpheme is *used* to express a position in an inflectional paradigm if and only if it is the closest one to that position in the geometrical space that we are exploring. In the examples from Nuer that follow, we will be able to look at a system with a great deal of syncretism, and see how our system works in that context.

The normalization of morpheme vectors—taking them to be of unit length—is both mathematically natural and linguistically significant. One of the central observations in inflectional morphology flows from this characteristic: the effect that is often referred to as *blocking*. It will frequently be found that one morpheme is used to express a part of a paradigm (for example, the verbal suffix -∅ marks present tense in weak verbs), and a different morpheme is used to express a subpart of that part (for example, *s* marks the 3rd person singular present). The morpheme that indicates the subpart takes precedence over the morpheme that indicates the larger domain. Thus, from the grammar's point of view, the null verbal suffix is not specified as *avoiding* the 3rd person singular in any sense, but it does not in fact appear in 3rd person singular forms because it is superseded by the more specific morpheme. This effect is a result of the geometry of the system. We have already seen this in the behavior just above of the suffix *-s* vis-a-vis the suffix -∅.[10]

Let's look at a few examples to illustrate these ideas.

1.7 German verb

Our first example will deal with the person and number suffixes for the German present tense verb. In this initial illustration, we will consider only the the suffix which marks person and number.

(18)

	NUMBER	
	singular	*plural*
PERSON		
1st	sing + e	sing + en
2nd	sing + st	sing + t
3rd	sing + t	sing + en

(19)

	e	st	en	t
present, 1, sg	1	-	-	-
present, 2, sg	-	1	-	-
present, 3, sg	-	-	-	1
present, 1, pl	-	-	1	-
present, 2, pl	-	-	-	1
present, 3, pl	-	-	1	-

[10] In section 3, we will return to the particular observation that in a broad range of cases, one finds a strong tendency for feature-values not to be realized on two different morphemes, intuitively speaking. In the framework we develop here, this is the result of a decomposition of the M vector into two parts that is effected by each morpheme realization.

The same matrices express the relationship between representations as we saw for the English weak verb, but here we have four suffixes instead of three: {-e, -st, -en, -t}. We follow the same steps as we did for English and begin by counting the occurrences of morphosyntactic features for each suffix; see (20), which is just like (16) above.

(20)

	e	st	en	t
past	0	0	0	0
present	1	1	2	2
1st	1	0	1	0
2nd	0	1	0	1
3rd	0	0	1	1
sg	1	1	0	1
pl	0	0	2	1

Again we normalize each column, giving us the following values.

(21)

	$\langle e\|$	$\langle st\|$	$\langle en\|$	$\langle t\|$
past	$\frac{0}{\sqrt{3}}$	$\frac{0}{\sqrt{3}}$	$\frac{0}{\sqrt{10}}$	$\frac{0}{\sqrt{8}}$
present	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{2}{\sqrt{10}}$	$\frac{1}{\sqrt{2}}$
1st	$\frac{1}{\sqrt{3}}$	$\frac{0}{\sqrt{3}}$	$\frac{1}{\sqrt{10}}$	$\frac{0}{\sqrt{8}}$
2nd	$\frac{0}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{0}{\sqrt{10}}$	$\frac{1}{\sqrt{9}}$
3rd	$\frac{0}{\sqrt{3}}$	$\frac{0}{\sqrt{3}}$	$\frac{1}{\sqrt{10}}$	$\frac{1}{\sqrt{8}}$
sg	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{0}{\sqrt{10}}$	$\frac{1}{\sqrt{8}}$
pl	$\frac{0}{\sqrt{3}}$	$\frac{0}{\sqrt{3}}$	$\frac{2}{\sqrt{10}}$	$\frac{1}{\sqrt{8}}$

Or numerically:

(22) $\mathcal{B}=$

	$\langle e\|$	$\langle st\|$	$\langle en\|$	$\langle t\|$
past	-	-	-	-
present	0.58	0.58	0.63	0.71
1	0.58	-	0.32	-
2	-	0.58	-	0.35
3	-	-	0.32	0.35
sg	0.58	0.58	-	0.35
pl	-	-	0.63	0.35

Limiting to the case of the present tense forms:

(23)

	present	1	2	3	sg	pl
present, 1, sg	1	1	-	-	1	-
present, 2, sg	1	-	1	-	1	-
present, 3, sg	1	-	-	1	1	-
present, 1, pl	1	1	-	-	-	1
present, 2, pl	1	-	1	-	-	1
present, 3, pl	1	-	-	1	-	1

Competition matrix:

(24) $\Phi\times\ \mathcal{B}=$

	e	st	en	t
present, 1, sg	1.73*	1.15	0.95	1.06
present, 2, sg	1.15	1.73*	0.63	1.41
present, 3, sg	1.15	1.15	0.95	1.41*
present, 1, pl	1.15	0.58	1.58*	1.06
present, 2, pl	0.58	1.15	1.26	1.41*
present, 3, pl	0.58	0.58	1.58*	1.41

This example is much like what we saw with the English weak verb, but here we have four suffixes instead of three: {-e, -st, -en, -t}. We follow the same steps as we did for English (what we call *smart initialization*, and begin by counting the occurrences of morphosyntactic features for each suffix; see (20), which is just like (16) above. We can see that for this paradigm, smart initialization resulted in the correct choice of affixes for each paradigm position. We next look at how smart initialization fares for Latin adjectives.

1.8 Latin adjectives

Here are the suffixes that occur on Latin adjectives according to their case, number and gender.

NUMBER	*Singular*			*Plural*		
GENDER	*masculine*	*feminine*	*neuter*	*masculine*	*feminine*	*neuter*
CASE:						
Nominative	us	a	um	i	ae	a
Genitive	i	ae	i	orum	arum	orum
Dative	o	ae	o	is	is	is
Accusative	um	am	um	os	as	a
Ablative	o	a	o	is	is	is
Vocative	e	a	um	i	ae	a

Table 2: Latin adjectives

When we apply smart initialization to this paradigm we get the following counts:

(25)

	us	i	o	um	e	a	ae	am	orum	is	as	arum	os
sg	1	2	4	4	1	3	2	1	0	0	0	0	0
pl	0	2	0	0	0	3	2	0	2	6	1	1	1
masc	1	3	2	1	1	0	0	0	1	2	0	0	1
fem	0	0	0	0	0	3	4	1	0	2	1	1	0
neu	0	1	2	3	0	3	0	0	1	2	0	0	0
nom	1	1	0	1	0	2	1	0	0	0	0	0	0
gen	0	2	0	0	0	0	1	0	2	0	0	1	0
dat	0	0	2	0	0	0	1	0	0	3	0	0	0
acc	0	0	0	2	0	1	0	1	0	0	1	0	1
abl	0	0	2	0	0	1	0	0	0	3	0	0	0
voc	0	1	0	1	1	2	1	0	0	0	0	0	0

Matrix $\mathcal{B}$ after normalizing:

(26) $\mathcal{B} =$

	us	i	o	um	e	a	ae	am	orum	is	as	arum	os
$\langle sg\vert$	0.577	0.408	0.707	0.707	0.577	0.442	0.378	0.577	0	0	0	0	0
$\langle pl\vert$	0	0.408	0	0	0	0.442	0.378	0	0.632	0.739	0.577	0.577	0.577
$\langle m\vert$	0.577	0.612	0.354	0.177	0.577	0	0	0	0.316	0.246	0	0	0.577
$\langle f\vert$	0	0	0	0	0	0.442	0.756	0.577	0	0.246	0.577	0.577	0
$\langle n\vert$	0	0.204	0.354	0.530	0	0.442	0	0	0.316	0.246	0	0	0
$\langle nom\vert$	0.577	0.204	0	0.177	0	0.295	0.189	0	0	0	0	0	0
$\langle gen\vert$	0	0.408	0	0	0	0	0.189	0	0.632	0	0	0.577	0
$\langle dat\vert$	0	0	0.354	0	0	0	0.189	0	0	0.369	0	0	0
$\langle acc\vert$	0	0	0	0.354	0	0.147	0	0.577	0	0	0.577	0	0.577
$\langle abl\vert$	0	0	0.354	0	0	0.147	0	0	0	0.369	0	0	0
$\langle voc\vert$	0	0.204	0	0.177	0.577	0.295	0.189	0	0	0	0	0	0

Of the thirty cases in the array, 26 are correct from smart initialization, and 4 are not correct, and must be specifically learned. These are the cases where the incorrect prediction is shown in red and the real winner in green. In §2 we shall look at a further step beyond smart initialization that enables a speaker to modify feature values in order to obtain the correct morpheme for each paradigm position in all cases.

(27) $\Phi \times \mathcal{B} =$

		us	i	o	um	e	a	ae	am	orum	is	as	arum	os
	n	1.732	1.225	1.061	1.061	1.155	0.737	0.567	0.577	0.316	0.246	0	0	0.577
	g	1.155	1.429	1.061	0.884	1.155	0.442	0.567	0.577	0.949	0.246	0	0.577	0.577
MSG	*d*	1.155	1.021	1.414	0.884	1.155	0.442	0.567	0.577	0.316	0.615	0	0	0.577
	ac	1.155	1.021	1.061	1.237	1.155	0.590	0.378	1.155	0.316	0.246	0.577	0	1.155
	ab	1.155	1.021	1.414	0.884	1.155	0.590	0.378	0.577	0.316	0.615	0	0	0.577
	v	1.155	1.225	1.061	1.061	1.732	0.737	0.567	0.577	0.316	0.246	0	0	0.577
	n	1.155	0.612	0.707	0.884	0.577	1.180	1.323	1.155	0	0.246	0.577	0.577	0
	g	0.577	0.816	0.707	0.707	0.577	0.885	1.323	1.155	0.632	0.246	0.577	1.155	0
FSG	*d*	0.577	0.408	1.061	0.707	0.577	0.885	1.323	1.155	0	0.615	0.577	0.577	0
	ac	0.577	0.408	0.707	1.061	0.577	1.032	1.134	1.732	0	0.246	1.155	0.577	0.577
	ab	0.577	0.408	1.061	0.707	0.577	1.032	1.134	1.155	0	0.615	0.577	0.577	0
	v	0.577	0.612	0.707	0.884	1.155	1.180	1.323	1.155	0	0.246	0.577	0.577	0
	n	1.155	0.816	1.061	1.414	0.577	1.180	0.567	0.577	0.316	0.246	0	0	0
	g	0.577	1.021	1.061	1.237	0.577	0.885	0.567	0.577	0.949	0.246	0	0.577	0
NSG	*d*	0.577	0.612	1.414	1.237	0.577	0.885	0.567	0.577	0.316	0.615	0	0	0
	ac	0.577	0.612	1.061	1.591	0.577	1.032	0.378	1.155	0.316	0.246	0.577	0	0.577
	ab	0.577	0.612	1.414	1.237	0.577	1.032	0.378	0.577	0.316	0.615	0	0	0
	v	0.577	0.816	1.061	1.414	1.155	1.180	0.567	0.577	0.316	0.246	0	0	0
	n	1.155	1.225	0.354	0.354	0.577	0.737	0.567	0	0.949	0.985	0.577	0.577	1.155
	g	0.577	1.429	0.354	0.177	0.577	0.442	0.567	0	1.581	0.985	0.577	1.155	1.155
MPL	*d*	0.577	1.021	0.707	0.177	0.577	0.442	0.567	0	0.949	1.354	0.577	0.577	1.155
	ac	0.577	1.021	0.354	0.530	0.577	0.590	0.378	0.577	0.949	0.985	1.155	0.577	1.732
	ab	0.577	1.021	0.707	0.177	0.577	0.590	0.378	0	0.949	1.354	0.577	0.577	1.155
	v	0.577	1.225	0.354	0.354	1.155	0.737	0.567	0	0.949	0.985	0.577	0.577	1.155
	n	0.577	0.612	0	0.177	0	1.180	1.323	0.577	0.632	0.985	1.155	1.155	0.577
	g	0	0.816	0	0	0	0.885	1.323	0.577	1.265	0.985	1.155	1.732	0.577
FPL	*d*	0	0.408	0.354	0	0	0.885	1.323	0.577	0.632	1.354	1.155	1.155	0.577
	ac	0	0.408	0	0.354	0	1.032	1.134	1.155	0.632	0.985	1.732	1.155	1.155
	ab	0	0.408	0.354	0	0	1.032	1.134	0.577	0.632	1.354	1.155	1.155	0.577
	v	0	0.612	0	0.177	0.577	1.180	1.323	0.577	0.632	0.985	1.155	1.155	0.577
	n	0.577	0.816	0.354	0.707	0	1.180	0.567	0	0.949	0.985	0.577	0.577	0.577
	g	0	1.021	0.354	0.530	0	0.885	0.567	0	1.581	0.985	0.577	1.155	0.577
NPL	*d*	0	0.612	0.707	0.530	0	0.885	0.567	0	0.949	1.354	0.577	0.577	0.577
	ac	0	0.612	0.354	0.884	0	1.032	0.378	0.577	0.949	0.985	1.155	0.577	1.155
	ab	0	0.612	0.707	0.530	0	1.032	0.378	0	0.949	1.354	0.577	0.577	0.577
	v	0	0.816	0.354	0.707	0.577	1.180	0.567	0	0.949	0.985	0.577	0.577	0.577

1.9 Russian, 1 class

The Russian nominal declension classes have been much studied in modern studies of inflectional morphology. Here we present the paradigm of a Russian noun that falls into the first of four classes.[11]

[11] The literature on Russian nominal inflectional morphology is large. See Corbett and Fraser (1993). Corbett and Fraser (1993, 114): "We have presented four declensional classes. This is not the traditional account; most descriptions recognize only three, treating zakon and v'ino as variants of a single declensional lcass (as in, for instance, Vinogradov et al. (1952), Unbegaun (1957) and Stankiewicz (1968).)"

12

(28)

	"law" CLASS I	
	singular	*plural*
nom	zakon	zakoni
gen	zakona	zakonov
acc	zakon	zakoni
loc	zakone	zakonax
dat	zakonu	zakonam
inst	zakonom	zakonam'i

Applying smart initialization gives us the following counts:

(29)

	∅	a	e	u	om	y	ov	ax	am	ami
sg	2	1	1	1	1	0	0	0	0	0
pl	0	0	0	0	0	2	1	1	1	1
nom	1	0	0	0	0	1	0	0	0	0
gen	0	1	0	0	0	0	1	0	0	0
acc	1	0	0	0	0	1	0	0	0	0
loc	0	0	1	0	0	0	0	1	0	0
dat	0	0	0	1	0	0	0	0	1	0
inst	0	0	0	0	1	0	0	0	0	1

Expressed algebraically:

(30) $$\mathcal{B} = \begin{bmatrix} \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{2}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{6}} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 \\ \frac{1}{\sqrt{6}} & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{6}} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} & 0 & 0 & 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Matrix $\mathcal{B}$ expressed numerically:

(31) $$\mathcal{B} = \begin{bmatrix} 0.816 & 0.707 & 0.707 & 0.707 & 0.707 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.816 & 0.707 & 0.707 & 0.707 & 0.707 \\ 0.408 & 0 & 0 & 0 & 0 & 0.408 & 0 & 0 & 0 & 0 \\ 0 & 0.707 & 0 & 0 & 0 & 0 & 0.707 & 0 & 0 & 0 \\ 0.408 & 0 & 0 & 0 & 0 & 0.408 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.707 & 0 & 0 & 0 & 0 & 0.707 & 0 & 0 \\ 0 & 0 & 0 & 0.707 & 0 & 0 & 0 & 0 & 0.707 & 0 \\ 0 & 0 & 0 & 0 & 0.707 & 0 & 0 & 0 & 0 & 0.707 \end{bmatrix}$$

[12] Regarding formalism, Corbett and Fraser (1993) reference Evans and Gazdar (1989), and also the ELU formalism of Russell et al. (1992) and Word Grammar formalism of Fraser and Hudson (1992).

We adopt the analysis by which [ɨ] is derived from more basic [i] when it follows a non-back hard consonant; some treat [ɨ] as a distinct underlying vowel. Consonants fall into three distinct categories with regarding to palatalization (the hard/soft contrast). Most have both a hard and soft version, which is indicated as C and C'. /č ž c/ are hard, and remain unchanged in environments where we would expect softening (notably before a suffix beginning with /e/). Two consonants, /č'/ and /šč'/, are soft in all positions (that is, morphemes containing these consonants appear in only one form). /k,g,x/ appear in soft form before /i/.

(32) $\phi \times \mathcal{B} =$

	$\emptyset$	*a*	*e*	*u*	*om*	*y*	*ov*	*ax*	*am*	*ami*
n.sg.	1.225	0.707	0.707	0.707	0.707	0.408	0	0	0	0
gen.sg.	0.816	1.414	0.707	0.707	0.707	0	0.707	0	0	0
acc.sg.	1.225	0.707	0.707	0.707	0.707	0.408	0	0	0	0
loc.sg.	0.816	0.707	1.414	0.707	0.707	0	0	0.707	0	0
dat.sg.	0.816	0.707	0.707	1.414	0.707	0	0	0	0.707	0
inst.sg.	0.816	0.707	0.707	0.707	1.414	0	0	0	0	0.707
nom.pl.	0.408	0	0	0	0	1.225	0.707	0.707	0.707	0.707
gen.pl.	0	0.707	0	0	0	0.816	1.414	0.707	0.707	0.707
acc.pl.	0.408	0	0	0	0	1.225	0.707	0.707	0.707	0.707
loc.pl.	0	0	0.707	0	0	0.816	0.707	1.414	0.707	0.707
dat.pl.	0	0	0	0.707	0	0.816	0.707	0.707	1.414	0.707
inst.pl.	0	0	0	0	0.707	0.816	0.707	0.707	0.707	1.414

We find that smart initialization works perfectly for this particular paradigm. We next look at what steps our model takes beyond smart initialization in order to deal with cases where it does not always predict the correct morpheme for a given paradigm position.

2 Learning: the Delta Rule

2.1 The role of learning in linguistic analysis

In this section, we will extend our concern with learning, which is to say, with construction of an algorithm that maps from data to the parametric values of the grammar.

Learning and learnability play an important role in the framework that we explore in this paper, and we would like to point out that in our view, the linguist should understand that success in understanding learning goes hand in hand with no longer accepting the idea that it is an *advantage* for a linguistic theory to rule out certain grammars, or that we should prefer theory A over theory B because the theory B permits some grammars or languages that theory A does not. To say that we do not accept those views seems like such heresy that we need to explain why a bit more. The reader is cetainly not obliged to accept any of our views on this subject, but it does inform the work that we present here.

There are four reasons why we *reject* the notion that the value of a linguistic proposal should be evaluated by how well it limits or restricts the class of possible human languages. The first is that the notion of "more restricted theory" was imported surreptiously into linguistics, inappropriately from Popper's conception of science and inappropriately as a place-holder for learnability. The second is that we have little evidence of what does not occur in grammar (either in occurring grammars or in non-occurring but possible grammars). The third (closely related to the second) is that we are still in an expansionary phase of linguistics, in which every successful piece of research involves the discovery of new organizational principles of grammar, and that no successful piece of research has ever succeeded by ruling out a set of grammars. The fourth is simply that the an implicit, and illicit, connection has been made between reducing the number of knowable human languages and the difficulty of solving the problem of language learning, but that implicit connection does not stand the light of what we know about machine learning today. We have added some remarks on this in an appendix at the end of this paper (secction 6).

2.2 Smart initialization: example from German

We have employed what we call *smart initialization* in the way we assigned the vectors associated with each morpheme. Smart initialization amounts to simply using the observed frequencies of feature values, observed in each of the cases where a particular morpheme is observed, to directly inform the coordinates of the morpheme. This method will often produce values that work correctly, but there is no guarantee that it will, and often enough it does not. We will explore here what learning needs to take place to correct the placement of the morphemes.[13]

Let's look at a case where smart initialization does not do the job. We looked at the case of person-number endings of the German verb in the present tense above, in section 1.7. If we add the past tense

[13] We would expect that smart initialization would constitute a diachronic attraction for a language, in the sense that all other things being equal, there would be a tendency for a language to shift towards the the system described by smart initialization.

forms of the weak verb in German, things do not work out as well for this method, and the 3rd person singular present is assigned the wrong suffix under smart initialization as we have presented it so far.[14] Consider the data in (33).

(33)

NUMBER	*singular*	*plural*
PERSON	*past*	
1st	lieb+t+e	lieb+t+en
2nd	lieb+t+(e)st	lieb+t+(e)t
3rd	lieb+t+e	lieb+t+en
	present	
1st	lieb+e	lieb+en
2nd	lieb+st	lieb+t
3rd	lieb+t	lieb+en

(34)

	e	st	en	t
past, 1, sg	1	-	-	-
past, 2, sg	-	1	-	-
past, 3, sg	1	-	-	-
past, 1, pl	-	-	1	-
past, 2, pl	-	-	-	1
past, 3, pl	-	-	1	-
present, 1, sg	1	-	-	-
present, 2, sg	-	1	-	-
present, 3, sg	-	-	-	1
present, 1, pl	-	-	1	-
present, 2, pl	-	-	-	1
present, 3, pl	-	-	1	-

(35) Φ=

	past	*present*	*1*	*2*	*3*	*sg*	*pl*
past, 1, sg	1	-	1	-	-	1	-
past, 2, sg	1	-	-	1	-	1	-
past, 3, sg	1	-	-	-	1	1	-
past, 1, pl	1	-	1	-	-	-	1
past, 2, pl	1	-	-	1	-	-	1
past, 3, pl	1	-	-	-	1	-	1
present, 1, sg	-	1	1	-	-	1	-
present, 2, sg	-	1	-	1	-	1	-
present, 3, sg	-	1	-	-	1	1	-
present, 1, pl	-	1	1	-	-	-	1
present, 2, pl	-	1	-	1	-	-	1
present, 3, pl	-	1	-	-	1	-	1

Count array:

(36)

	e	st	en	t
past	2	1	2	1
present	1	1	2	2
1	2	-	2	-
2	-	2	-	2
3	1	-	2	1
sg	3	2	-	1
pl	-	-	4	2

[14] A weak verb is one whose stem does not change in the past, and which contains a past tense suffix -t- between the stem and the person-number suffix. The account is due to Jacob Grimm 1819.

(37)

	$\langle e\|$	$\langle st\|$	$\langle en\|$	$\langle t\|$
past	$\frac{2}{\sqrt{19}}$	$\frac{1}{\sqrt{10}}$	$\frac{2}{4\sqrt{2}}$	$\frac{1}{\sqrt{15}}$
present	$\frac{1}{\sqrt{19}}$	$\frac{1}{\sqrt{10}}$	$\frac{2}{4\sqrt{2}}$	$\frac{2}{\sqrt{15}}$
1st	$\frac{2}{\sqrt{19}}$	$\frac{0}{\sqrt{10}}$	$\frac{2}{4\sqrt{2}}$	$\frac{0}{\sqrt{15}}$
2nd	$\frac{0}{\sqrt{19}}$	$\frac{2}{\sqrt{10}}$	$\frac{0}{4\sqrt{2}}$	$\frac{2}{\sqrt{15}}$
3rd	$\frac{1}{\sqrt{19}}$	$\frac{0}{\sqrt{10}}$	$\frac{2}{4\sqrt{2}}$	$\frac{1}{\sqrt{15}}$
sg	$\frac{3}{\sqrt{19}}$	$\frac{2}{\sqrt{10}}$	$\frac{0}{4\sqrt{2}}$	$\frac{1}{\sqrt{15}}$
pl	$\frac{0}{\sqrt{19}}$	$\frac{0}{\sqrt{10}}$	$\frac{4}{4\sqrt{2}}$	$\frac{2}{\sqrt{15}}$

$= \mathcal{B}.$

(38)

	e	st	en	t
past	0.46	0.32	0.35	0.26
present	0.23	0.32	0.35	0.52
1	0.46	-	0.35	-
2	-	0.63	-	0.52
3	0.23	-	0.35	0.26
sg	0.69	0.63	-	0.26
pl	-	-	0.71	0.52

$= \mathcal{B}$

$\Phi \times \mathcal{B}$: Competition matrix

(39)

	e	st	en	t
past, 1, sg	1.61*	0.95	0.71	0.52
past, 2, sg	1.15	1.58*	0.35	1.03
past, 3, sg	1.38*	0.95	0.71	0.77
past, 1, pl	0.92	0.32	1.41*	0.77
past, 2, pl	0.46	0.95	1.06	1.29*
past, 3, pl	0.69	0.32	1.41*	1.03
present, 1, sg	1.38*	0.95	0.71	0.77
present, 2, sg	0.92	1.58*	0.35	1.29
present, 3, sg	1.15*	0.95	0.71	1.03
present, 1, pl	0.69	0.32	1.41*	1.03
present, 2, pl	0.23	0.95	1.06	1.55*
present, 3, pl	0.46	0.32	1.41*	1.29

(40) $\mathcal{B} = \begin{bmatrix} 0.459 & 0.229 & 0.459 & 0 & 0.229 & 0.688 & 0 \\ 0.316 & 0.316 & 0 & 0.632 & 0 & 0.632 & 0 \\ 0.354 & 0.354 & 0.354 & 0 & 0.354 & 0 & 0.707 \\ 0.258 & 0.516 & 0 & 0.516 & 0.258 & 0.258 & 0.516 \end{bmatrix}$

We repeat here the values of $\Phi \times \mathcal{B}$ in a table, with the maximum values for each row in blue.

(41) $\Phi \times \mathcal{B}=$

suffix	-e	-st	-en	-t
1st sg past	1.606	0.949	0.707	0.516
2nd sg past	1.147	1.581	0.354	1.033
3rd sg past	1.376	0.949	0.707	0.775
1st pl past	0.918	0.316	1.414	0.775
2nd pl past	0.459	0.949	1.061	1.291
3rd pl past	0.688	0.316	1.414	1.033
1st sg pres	1.376	0.949	0.707	0.775
2nd sg pres	0.918	1.581	0.354	1.291
3rd sg pres	1.147	0.949	0.707	1.033
1st pl pres	0.688	0.316	1.414	1.033
2nd pl pres	0.229	0.949	1.061	1.549
3rd pl pres	0.459	0.316	1.414	1.291

All the paradigm positions have the maximum value on the correct morpheme except the 3rd singular present, where the incorrect winner is marked in red. The intended winner, *-t* loses by 0.114.

(42) $\Phi \times \mathcal{B}=$

suffix	-e	-st	-en	-t
1st sg past	1.606	0.949	0.707	0.516
2nd sg past	1.147	1.581	0.354	1.033
3rd sg past	1.376	0.949	0.707	0.775
1st pl past	0.918	0.316	1.414	0.775
2nd pl past	0.459	0.949	1.061	1.291
3rd pl past	0.688	0.316	1.414	1.033
1st sg pres	1.376	0.949	0.707	0.775
2nd sg pres	0.918	1.581	0.354	1.291
3rd sg pres	1.147	0.949	0.707	1.033
1st pl pres	0.688	0.316	1.414	1.033
2nd pl pres	0.229	0.949	1.061	1.549
3rd pl pres	0.459	0.316	1.414	1.291

The winner is *e* (shown in red) whereas the correct suffix is *t* (shown in green). What does it mean that the system did not obtain the correct results, even though in some sense we fed it all the right answers?

Let's address a question about the learning of inflectional morphology that is quite basic, and has an impact on how we analyze anybody's theory and their account of any particular language. When we analyze a morphology, we typically do it with a full set of specifications of the paradigms of the language. Given a full set of paradigms, along with one or two examples for each grouping, we expect of our analysis that it can extend the correct results to unseen words. But we also typically expect that the analysis of the data that we are given will be simpler than (merely) repeating all of the given data. To take a simple case, when we analyze the present tense of the verb in English, we do not specify the form for each of the six person/number combinations; we typically say what the 3rd sg present form is, and then say that all of the other present forms take a different form.

But an account of morphology that takes learning seriously must also do something else: it must be able to provide an analysis of an *incomplete* set of data, making predictions about the forms that it has not yet been given. The more successful a theory is in this regard — that is, in inferring the correct answers on the basis of incomplete data—the better the theory is.

2.2.1 Delta rule

When there is a lack of agreement between the ceorrect choice of affix and the predicted choice of affix (and at this point, such a lack of agreement must be the result of our so-called smart initialization not being quite smart enough!), we want our system to be modified automatically in order to no longer make that error.

Weights that will determine the correct suffix for every feature-value combination can be learned through a simple gradient descent algorithm, known as the Delta Rule, which goes back to the perceptron learning rule. The intuition that lies behind it is that if we can identify an input that leads to an incorrect output (input i to output j, let's say), we should change the weight from i to j in proportion to the strength of the signal that went to i and in proportion to the difference between the calculated output and the desired output.

We choose a small stepsize η for each iteration of the algorithm in which we modify each suffix vector μ_j according to the following formula.

(43) $\langle \mu_j | := \langle \mu_j | - \eta(\hat{\mu_j} - \Phi_i \langle \mu_j |)(-\Phi_i)$

where μ_j is the j^{th} suffix, $\langle \mu_j |$ is its representation in feature-value space (here, R^7), $\hat{\mu}$ is the suffix's paradigm representation (here, in R^{12}).[15] After each modification of μ_j, the vector is re-normalized.

2.3 German

Applying the Delta Rule to the German verb suffix vectors, with a stepsize $\eta = 0.1$ and modifying the vectors only for paradigm positions that have the incorrect winner, we get correct values after just one iteration, as shown below.

(44) $\mu =$

$$\begin{bmatrix} 0.521 & 0.344 & 0.370 & 0.259 \\ 0.130 & 0.241 & 0.296 & 0.515 \\ 0.521 & 0 & 0.370 & 0 \\ 0 & 0.687 & 0 & 0.518 \\ 0.130 & -0.103 & 0.296 & 0.256 \\ 0.651 & 0.584 & -0.074 & 0.256 \\ 0 & 0 & 0.739 & 0.518 \end{bmatrix}$$

(45) $\Phi \times \mu =$

	$-e$	$-st$	$-en$	$-t$
1st sg pst	1.692	0.928	0.665	0.515
2nd sg pst	1.172	1.615	0.296	1.033
3rd sg pst	1.302	0.825	0.591	0.771
1st pl pst	1.042	0.344	1.478	0.777
2nd pl pst	0.521	1.031	1.109	1.295
3rd pl pst	0.651	0.241	1.405	1.033
1st sg pres	1.302	0.825	0.591	0.771
2nd sg pres	0.781	1.512	0.222	1.289
3rd sg pres	0.911	0.722	0.517	1.026
1st pl pres	0.651	0.241	1.405	1.033
2nd pl pres	0.130	0.928	1.035	1.551
3rd pl pres	0.260	0.137	1.331	1.289

[15] The rule can be derived by calculating the partial derivative of the loss function $L = \frac{1}{2}(\hat{\mu j} - \Phi_i \mu_j)^2$ with respect to μ_j. A modified version of this algorithm will update a vector μ_j only when it is the incorrect choice for a given feature combination.

2.4 Latin

The values of $\Phi \times \mathcal{B}$ for the Latin adjective are repeated below from (27) as (46).

(46) $\Phi \times \mathcal{B} =$

		us	i	o	um	e	a	ae	am	orum	is	as	arum	os
	n	1.732	1.225	1.061	1.061	1.155	0.737	0.567	0.577	0.316	0.246	0	0	0.577
	g	1.155	1.429	1.061	0.884	1.155	0.442	0.567	0.577	0.949	0.246	0	0.577	0.577
MSG	d	1.155	1.021	1.414	0.884	1.155	0.442	0.567	0.577	0.316	0.615	0	0	0.577
	ac	1.155	1.021	1.061	1.237	1.155	0.590	0.378	1.155	0.316	0.246	0.577	0	1.155
	ab	1.155	1.021	1.414	0.884	1.155	0.590	0.378	0.577	0.316	0.615	0	0	0.577
	v	1.155	1.225	1.061	1.061	1.732	0.737	0.567	0.577	0.316	0.246	0	0	0.577
	n	1.155	0.612	0.707	0.884	0.577	1.180	1.323	1.155	0	0.246	0.577	0.577	0
	g	0.577	0.816	0.707	0.707	0.577	0.885	1.323	1.155	0.632	0.246	0.577	1.155	0
FSG	d	0.577	0.408	1.061	0.707	0.577	0.885	1.323	1.155	0	0.615	0.577	0.577	0
	ac	0.577	0.408	0.707	1.061	0.577	1.032	1.134	1.732	0	0.246	1.155	0.577	0.577
	ab	0.577	0.408	1.061	0.707	0.577	1.032	1.134	1.155	0	0.615	0.577	0.577	0
	v	0.577	0.612	0.707	0.884	1.155	1.180	1.323	1.155	0	0.246	0.577	0.577	0
	n	1.155	0.816	1.061	1.414	0.577	1.180	0.567	0.577	0.316	0.246	0	0	0
	g	0.577	1.021	1.061	1.237	0.577	0.885	0.567	0.577	0.949	0.246	0	0.577	0
NSG	d	0.577	0.612	1.414	1.237	0.577	0.885	0.567	0.577	0.316	0.615	0	0	0
	ac	0.577	0.612	1.061	1.591	0.577	1.032	0.378	1.155	0.316	0.246	0.577	0	0.577
	ab	0.577	0.612	1.414	1.237	0.577	1.032	0.378	0.577	0.316	0.615	0	0	0
	v	0.577	0.816	1.061	1.414	1.155	1.180	0.567	0.577	0.316	0.246	0	0	0
	n	1.155	1.225	0.354	0.354	0.577	0.737	0.567	0	0.949	0.985	0.577	0.577	1.155
	g	0.577	1.429	0.354	0.177	0.577	0.442	0.567	0	1.581	0.985	0.577	1.155	1.155
MPL	d	0.577	1.021	0.707	0.177	0.577	0.442	0.567	0	0.949	1.354	0.577	0.577	1.155
	ac	0.577	1.021	0.354	0.530	0.577	0.590	0.378	0.577	0.949	0.985	1.155	0.577	1.732
	ab	0.577	1.021	0.707	0.177	0.577	0.590	0.378	0	0.949	1.354	0.577	0.577	1.155
	v	0.577	1.225	0.354	0.354	1.155	0.737	0.567	0	0.949	0.985	0.577	0.577	1.155
	n	0.577	0.612	0	0.177	0	1.180	1.323	0.577	0.632	0.985	1.155	1.155	0.577
	g	0	0.816	0	0	0	0.885	1.323	0.577	1.265	0.985	1.155	1.732	0.577
FPL	d	0	0.408	0.354	0	0	0.885	1.323	0.577	0.632	1.354	1.155	1.155	0.577
	ac	0	0.408	0	0.354	0	1.032	1.134	1.155	0.632	0.985	1.732	1.155	1.155
	ab	0	0.408	0.354	0	0	1.032	1.134	0.577	0.632	1.354	1.155	1.155	0.577
	v	0	0.612	0	0.177	0.577	1.180	1.323	0.577	0.632	0.985	1.155	1.155	0.577
	n	0.577	0.816	0.354	0.707	0	1.180	0.567	0	0.949	0.985	0.577	0.577	0.577
	g	0	1.021	0.354	0.530	0	0.885	0.567	0	1.581	0.985	0.577	1.155	0.577
NPL	d	0	0.612	0.707	0.530	0	0.885	0.567	0	0.949	1.354	0.577	0.577	0.577
	ac	0	0.612	0.354	0.884	0	1.032	0.378	0.577	0.949	0.985	1.155	0.577	1.155
	ab	0	0.612	0.707	0.530	0	1.032	0.378	0	0.949	1.354	0.577	0.577	0.577
	v	0	0.816	0.354	0.707	0.577	1.180	0.567	0	0.949	0.985	0.577	0.577	0.577

Recall that there were four positions (shown above in red) in which smart initialization chose the wrong form.

In six iterations of the Delta Rule with stepsize $\eta = 0.1$ we end up with the correct forms for all 36 positions. Here are the final values of suffix vectors:

(47) $\mathcal{B} =$

us	i	o	um	e	a	ae	am	orum	is	as	arum	os
0.48	0.33	0.44	0.59	0.39	0.36	0.10	0.31	−0.25	−0.16	−0.23	−0.27	−0.16
−0.26	0.20	−0.14	−0.19	−0.17	0.28	0.23	−0.11	0.40	0.51	0.30	0.29	0.21
0.54	0.59	0.35	0.25	0.52	−0.25	−0.25	−0.02	0.18	0.04	−0.19	−0.21	0.47
−0.23	−0.26	−0.32	−0.27	−0.23	0.43	0.78	0.45	−0.12	0.18	0.51	0.44	−0.10
−0.08	0.20	0.27	0.43	−0.06	0.45	−0.20	−0.23	0.08	0.14	−0.24	−0.21	−0.32
0.55	0.19	−0.15	0.11	−0.21	0.27	0.03	−0.13	−0.21	−0.24	−0.16	−0.18	−0.30
−0.15	0.48	−0.21	−0.28	−0.14	−0.19	0.35	−0.05	0.79	−0.09	0.04	0.67	−0.06
0.10	−0.05	0.50	0.10	0.10	−0.18	0.09	−0.08	−0.12	0.52	−0.24	−0.23	−0.10
−0.10	−0.27	−0.13	0.39	−0.08	0.18	−0.07	0.75	−0.19	−0.22	0.63	−0.05	0.68
−0.07	−0.05	0.39	−0.10	−0.08	0.22	−0.23	−0.17	0	0.51	−0.08	−0.07	0
−0.12	0.23	−0.09	0.18	0.64	0.33	0.17	−0.12	−0.12	−0.13	−0.12	−0.12	−0.17

And here are the final values for suffixes projected onto paradigm positions.

(48) $\Phi \times \mathcal{B} =$

us	i	o	um	e	a	ae	am	orum	is	as	arum	os
1.56	1.10	0.64	0.95	0.70	0.38	−0.12	0.15	−0.28	−0.36	−0.58	−0.65	0.02
0.87	1.40	0.58	0.56	0.78	−0.09	0.20	0.23	0.72	−0.21	−0.38	0.20	0.25
1.11	0.87	1.29	0.94	1.01	−0.08	−0.05	0.20	−0.18	0.41	−0.66	−0.70	0.22
0.92	0.65	0.65	1.23	0.83	0.29	−0.22	1.04	−0.26	−0.34	0.21	−0.53	1.00
0.95	0.87	1.17	0.74	0.84	0.33	−0.38	0.12	−0.07	0.39	−0.50	−0.55	0.31
0.90	1.14	0.70	1.02	1.55	0.44	0.03	0.17	−0.19	−0.25	−0.54	−0.59	0.14
0.79	0.26	−0.02	0.43	−0.05	1.06	0.91	0.63	−0.58	−0.22	0.11	−0.01	−0.55
0.10	0.55	−0.09	0.04	0.03	0.59	1.23	0.71	0.42	−0.07	0.32	0.85	−0.32
0.34	0.02	0.63	0.42	0.26	0.60	0.98	0.68	−0.48	0.55	0.04	−0.05	−0.35
0.15	−0.20	−0.01	0.71	0.08	0.97	0.81	1.51	−0.56	−0.20	0.91	0.12	0.43
0.18	0.03	0.51	0.22	0.09	1.01	0.65	0.59	−0.37	0.53	0.20	0.10	−0.25
0.13	0.30	0.03	0.50	0.80	1.12	1.06	0.65	−0.49	−0.11	0.16	0.06	−0.43
0.95	0.71	0.56	1.14	0.12	1.08	−0.07	−0.05	−0.39	−0.26	−0.63	−0.66	−0.78
0.25	1.01	0.50	0.75	0.19	0.62	0.24	0.03	0.61	−0.11	−0.43	0.19	−0.54
0.50	0.48	1.21	1.12	0.42	0.63	−0.01	−0.00	−0.29	0.51	−0.71	−0.71	−0.58
0.30	0.26	0.58	1.41	0.25	0.99	−0.18	0.83	−0.36	−0.24	0.17	−0.53	0.20
0.33	0.48	1.10	0.93	0.25	1.03	−0.33	−0.09	−0.17	0.49	−0.55	−0.55	−0.48
0.28	0.75	0.62	1.21	0.96	1.14	0.07	−0.04	−0.29	−0.15	−0.59	−0.60	−0.65
0.83	0.98	0.06	0.17	0.14	0.30	0.02	−0.27	0.37	0.30	−0.06	−0.09	0.38
0.13	1.28	−0.00	−0.22	0.22	−0.16	0.34	−0.19	1.37	0.45	0.15	0.76	0.61
0.38	0.75	0.71	0.16	0.45	−0.15	0.08	−0.22	0.47	1.07	−0.13	−0.14	0.58
0.18	0.52	0.07	0.45	0.27	0.21	−0.09	0.62	0.39	0.33	0.74	0.04	1.36
0.22	0.75	0.59	−0.04	0.28	0.25	−0.24	−0.30	0.58	1.06	0.03	0.02	0.68
0.16	1.02	0.12	0.24	0.99	0.36	0.16	−0.25	0.46	0.42	−0.02	−0.03	0.50
0.06	0.13	−0.60	−0.34	−0.61	0.98	1.05	0.21	0.07	0.44	0.64	0.56	−0.19
−0.64	0.43	−0.67	−0.73	−0.53	0.52	1.36	0.29	1.07	0.59	0.84	1.41	0.05
−0.39	−0.10	0.05	−0.36	−0.30	0.53	1.11	0.26	0.17	1.21	0.57	0.51	0.01
−0.59	−0.33	−0.59	−0.07	−0.48	0.89	0.94	1.10	0.09	0.47	1.44	0.69	0.79
−0.55	−0.10	−0.07	−0.55	−0.47	0.93	0.79	0.18	0.28	1.19	0.73	0.67	0.11
−0.61	0.17	−0.55	−0.27	0.24	1.05	1.19	0.23	0.16	0.55	0.68	0.62	−0.07
0.21	0.59	−0.02	0.36	−0.44	1.01	0.06	−0.47	0.26	0.40	−0.11	−0.09	−0.41
−0.48	0.88	−0.08	−0.03	−0.37	0.54	0.38	−0.39	1.26	0.55	0.10	0.76	−0.18
−0.23	0.35	0.63	0.35	−0.13	0.55	0.12	−0.42	0.36	1.17	−0.18	−0.14	−0.21
−0.43	0.13	−0.00	0.64	−0.31	0.92	−0.05	0.41	0.29	0.43	0.69	0.03	0.57
−0.40	0.36	0.52	0.15	−0.31	0.95	−0.20	−0.51	0.48	1.15	−0.02	0.01	−0.12
−0.45	0.63	0.04	0.43	0.40	1.07	0.20	−0.46	0.36	0.52	−0.06	−0.03	−0.29

To summarize this section: the Delta Rule is an error-driven machine learning tool that enables us to simulate the way a learner might proceed beyond smart initialization of feature values for a set of exponents in a paradigm to arrive at a set of values that will find the correct morpheme for each

morphosyntactic position. We next look at ways in which our model can deal with the way stems and affixes interact in cases where a learner needs to select both the correct stem and the correct affix for a given paradigm position.

3 Morpheme concatenation as vector sum

We have so far considered only the selection of a single morpheme in all of our computations, but in the more general case, we wish to be able to select multiple morphemes. For example, if a stem morpheme has two allomorphs, and a suffix must also be chosen, then it is a *pair*, the stem and suffix, which must be selected. The central notion of geometrical morphology provides a prediction for these cases: the (stem, affix) pair provides a vector sum (the vector sum of the vector representing the stem and the vector representing the affix), and that pair is chosen whose vector sum is closest to the target position M. If a stem μ_t is realized from a set of allomorphs $\mathcal{T}$ and its suffix is selected from a set of inflectional suffixes $\mathcal{F}$, then to realize the position M in its inflectional paradigm, the morphology selects one stem allomorph $\hat{\mu}_{\text{stem}}$ in $\mathcal{T}$ and one affix $\hat{\mu}_{\text{affix}}$ in $\mathcal{F}$:

$$(\hat{\mu}_{\text{stem}}, \hat{\mu}_{\text{affix}}) = argmin_{\mu_{\text{stem}} \in \mathcal{T}, \mu_{\text{affix}} \in \mathcal{F}}\, distance(M, \vec{\mu}_{\text{stem}} + \vec{\mu}_{\text{affix}})$$

3.1 German plurals

We shall first illustrate this with one of the simplest possible examples: the plural suffix in German. German has a number of different plural suffixes whose choice depends on the stem with which the suffix occurs. The singular is unmarked, so we shall continue to indicate that suffix as $\emptyset$, remaining agnostic about whether this is a null morpheme or simply the lack of a morpheme. The following are some examples of plural suffixes in German.[16]

Noun	Pl. suffix	Pl. form	Gloss
Kind	er	Kinder	'child'
Glas	[−*back*] er	Gläser	'glass'
Fenster	$\emptyset$	Fenster	'window'
Mutter	[−*back*]	Mütter	'mother'
Auto	s	Autos	'automobile'

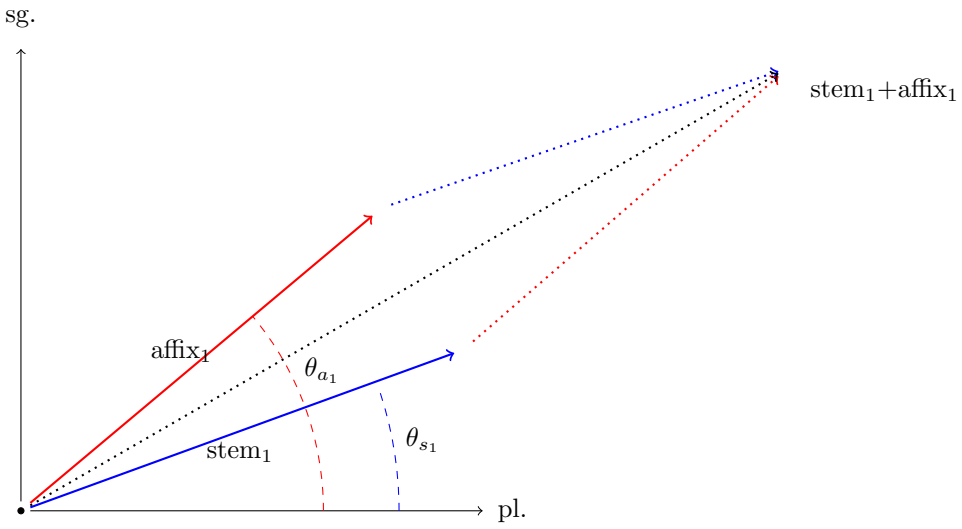
Table 3: Some German plurals

We only have two feature values in feature-value space: singular and plural. In that space are two-dimensional vectors for all the stems $\mathcal{T}$ and affixes $\mathcal{F}$. If we visualize the singular dimension as the y-axis and the plural dimension as the x-axis, given that we are requiring all the vectors to be of unit length, we represent each one simply by an angle θ_i: the angle between the position of the vector and the plural axis. For a given choice of noun stem, for the plural, the affix with which it occurs will be the one for which the vector sum of the stem with affix makes the smallest angle with the plural axis: i.e. has the smallest angular measure in the way we are assigning angular measures to vectors. Similarly, the affix that occurs with a given stem on the singular will be the one for which the angle of the stem plus affix is closest to $\frac{\pi}{2}$ or 90^o.

As the following diagram shows, the angle subtended by the vector sum of stem + affix is $\frac{\theta_{stem}+\theta_{affix}}{2}$ The angle between the thick blue and red vectors, stem$_1$ and affix$_1$ is $\theta_{a1} - \theta_{s1}$ so the angle below the black dotted line from the origin to the sum point stem$_1$ + affix$_1$ is $\theta_{s1} + \frac{\theta_{a1}-\theta_{s1}}{2} = \frac{\theta_{s1}+\theta_{a1}}{2}$.

[16] Where umlaut is part of the suffix, we represent umlaut as a floating [−*back*] feature and abstract away here from the question of exactly how and where this feature is realized.

(49)



Our model can learn feature values for stems and affixes so that the correct affixes will occur with the correct stems through the following algorithm.

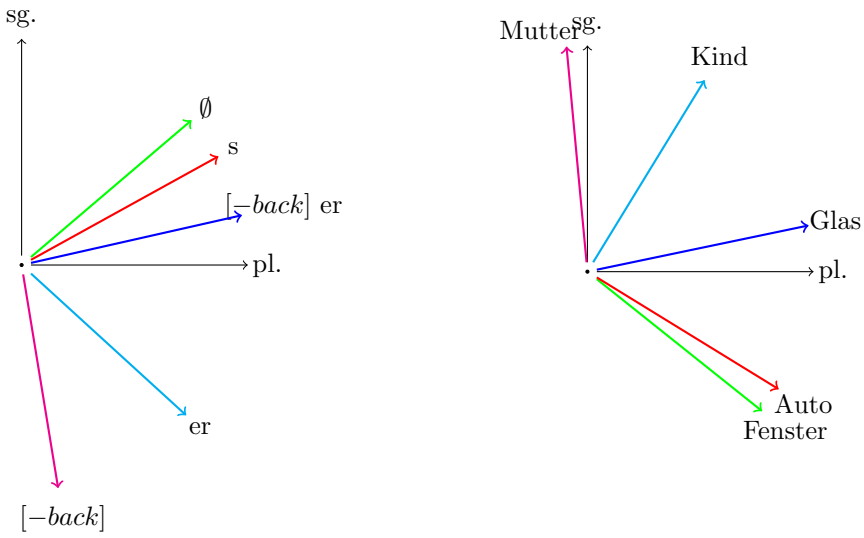
1. Initialize feature values for the stems and affixes as randomly chosen angles between $\frac{pi}{2}$ (90^o) and $-\frac{pi}{2}$ (-90^o).
2. Set some small stepsize such as 0.01 radians and a small margin of required separation such as 0.05 radians.
3. Repeat each of the following steps for each of a series of iterations until no adjustments need to be made.
   (a) For each stem:
      i. For each affix:
         A. If that combination is correct for the singular: if any other affix with the same stem results in an angle closer to 90^o than this combination does, move both the stem and the correct affix closer to 90^o and the incorrect one farther from 90^o, each by the stepsize.
         B. Do likewise for the plural with respect to the angle 0^o.

With a stepsize of $\eta = 0.01$ and margin $\epsilon = 0.05$ one run found, after 22 iterations, the following feature values expressed as angles in degrees from the plural axis.

(50)

Stem or affix	Angle	Plural FV	Singular FV
Fenster	-38.593	0.782	-0.624
Auto	-31.568	0.852	-0.524
Glas	11.800	0.979	0.204
Kind	58.535	0.522	0.853
Mutter	95.310	-0.093	0.996
$\emptyset$	40.430	0.761	0.649
s	28.909	0.875	0.483
[$-back$] er	12.729	0.975	0.220
er	-42.441	0.738	-0.675
[$-back$]	-80.694	0.162	-0.987

(51)

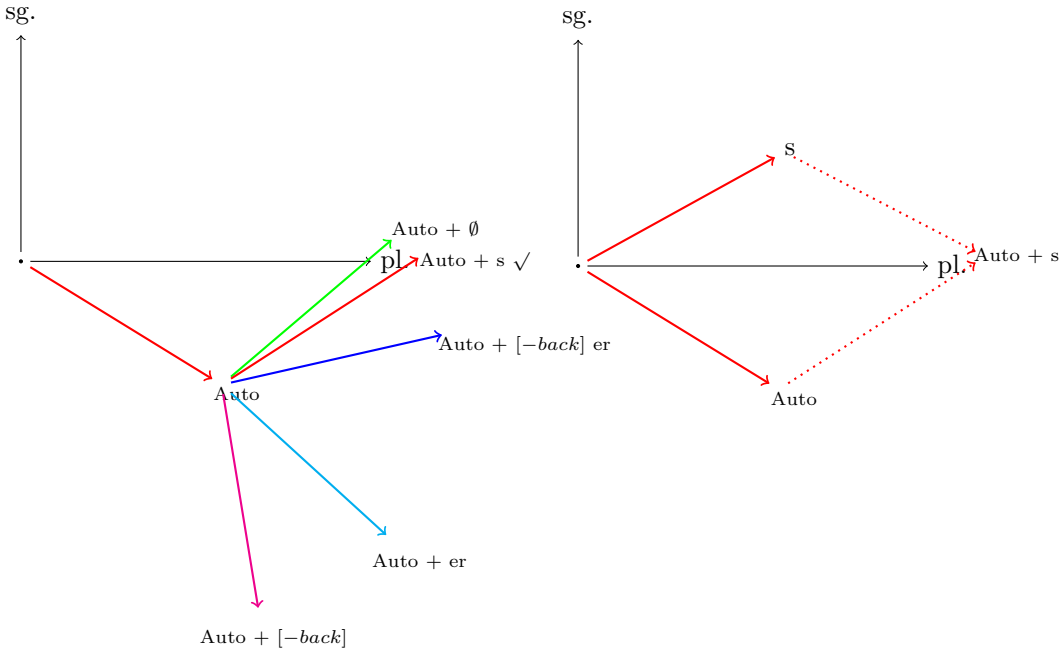


It is noteworthy that the vector for ∅ has ended up closer to the singular axis than any other affix. This ensures that it will be the affix that occurs with each stem in the singular. And because stem *Fenster* is the only one that occurs with the affix ∅ in the plural as well as the singular, it must have the lowest (in this case negative) value for singular among all the stems so that when combined with affix ∅ it will be closest to the plural axis than its combination with any other affixes.

It is important to note that when stems come from different lexemes, as is the case here, we take it that the learner has other ways of determining which lexeme (and therefore which stem) is the one in question. It will be the stem the chooses the correct affix for a given morphosyntactic combination; not the affix that chooses the stem.

For ease of exposition, we show how just one stem chooses the correct affix for plural. We can see that the vector sum *Auto* + *s* comes closest to having its endpoint on the plural axis. The diagram on the right shows more precisely how the two vectors *Auto* and *s* add up.

(52)



3.2 Spanish verbal classes

Spanish, like other Romance languages, is often treated as having three inflectional classes for verbs, labelled by their infinitival suffix: -ar verbs, -er verbs, and -ir verbs. The three different vowels of

these suffixes may well be analyzed as a *theme vowel*, a morpheme separate from the root and from the inflectional affixes. But for purposes of presentation, we will analyze Spanish more simply, and assume that there are three distinct inflectional classes among the verbs, and each class selects its own set of inflectional suffixes.

(53)

Class -ar infinitive:	sing cantar	call llamar
1st sg.	canto	llamo
2nd sg.	cantas	llamas
3rd sg.	canta	llama
1st pl.	cantamos	llamamos
2nd pl.	cantáis	llamáis
3rd pl.	cantan	llaman

Class -er infinitive	eat comer	fear temer
1st sg.	como	temo
2nd sg.	comes	temes
3rd sg.	come	teme
1st pl.	comemos	tememos
2nd pl.	coméis	teméis
3rd pl.	comen	temen

Class -ir infinitive	open abrir	live vivir
1st sg.	abro	vivo
2nd sg.	abres	vives
3rd sg.	abre	vive
1st pl.	abrimos	vivimos
2nd pl.	abrís	vivís
3rd pl.	abren	viven

From these data we can abstract the following inflectional classes of suffixes:

(54)

Class 1	
infinitive	ar
1st sg.	o
2nd sg.	as
3rd sg.	a
1st pl.	amos
2nd pl.	áis
3rd pl.	an

Class 2	
infinitive	er
1st sg.	o
2nd sg.	es
3rd sg.	e
1st pl.	emos
2nd pl.	éis
3rd pl.	en

Class 3	
infinitive	ir
1st sg.	o
2nd sg.	es
3rd sg.	e
1st pl.	imos
2nd pl.	ís
3rd pl.	en

Let's consider the selection of the morpheme *-as* for the *ar* Class 2nd person singular form, and of *-es* for the *-er* Class 2nd person singular form. If we repeat the process we have used so far, our "smart initialization," we will continue as in (55):

(55)

	-o	-as	-es
present, 1, sg	1	-	-
present, 2, sg	-	1	1
present, 3, sg	-	-	-
present, 1, pl	-	-	-
present, 2, pl	-	-	-
present, 3, pl	-	-	-

	-o	-as	-es
present	1	1	1
1st	1	0	0
2nd	0	1	1
3rd	0	0	0
sg	1	1	1
pl	0	0	0

Suffixes of different inflectional patterns will generally be placed in the same position in FV space by smart initialization, but they in fact appear in complementary distribution with each other. It must be, therefore, that the suffixes are *not* in the same position, just as an -ar stem must be in a different position from an -er stem. Consider the positioning as in $\mathcal{B}$ in (55)

(56)

	-o	-as	-es	$cant_{ar}$	com_{er}
present	1	1	1	1	1
1st	1	0	0		
2nd	0	1	1		
3rd	0	0	0		
sg	1	1	1		
pl	0	0	0		

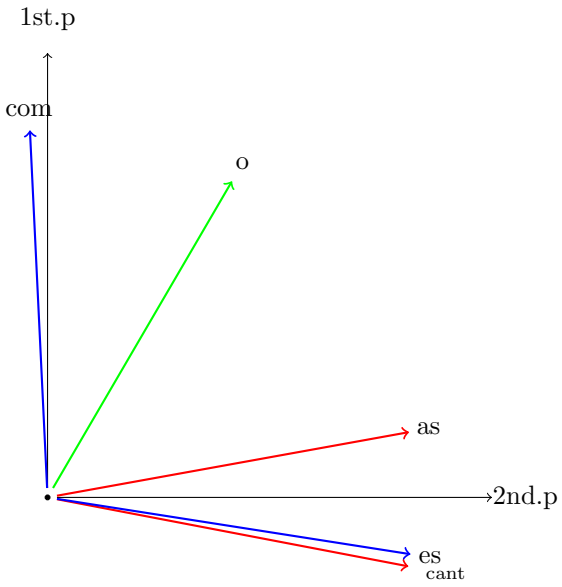
This initialization has given the same feature values to the two stems and also the same values to the affixes *-as* and *-es*. Consider the following configuration in which the vectors for the stems and for the two 2nd sg. suffixes are pulled apart.

Values of morphemes in radians	
cant	-0.18875
com	1.6188
-o	1.04273
-as	0.17836
-es	- 0.15520

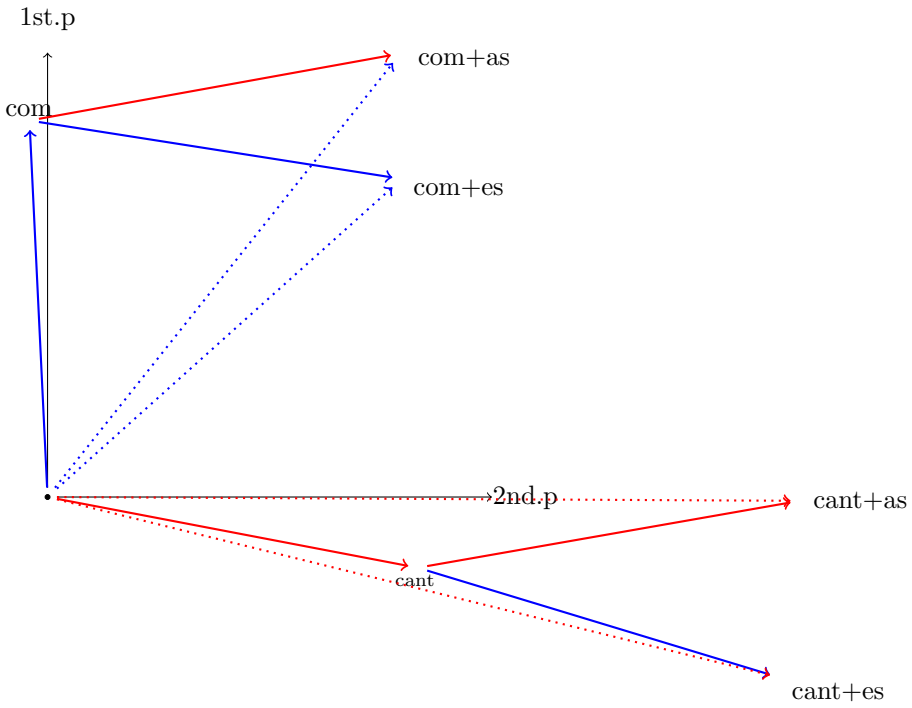
Table 4: Some possible values for Spanish singular present

The following graph shows how these values are located in the two-dimensional subspace of first and second person feature values. We can see that the *o* suffix will occur with either stem for the first person since it is the closest to that axis.

(57)



Here is how stem *cant* chooses affix *-as* over *-es* and stem *com* chooses affix *-es* over *as* in the second person. The vector sum of *com+es* is closer to the 2nd person axis than *com+as* and the vector sum of *cant+as* is closer to the second person axis than *cant+es*.



(58)

4 Multiple patterns of inflection within a language

4.1 General discussion

A central part of the task of analyzing inflectional morphologies is the analysis of different inflectional classes within a single language. That is, it is often the case that the specific choice of inflectional affixes is not fixed once and for all for all lexical stems in a given category, but rather falls into as many as several dozen patterns. These patterns often show striking similarities, all the while maintaining their differences. We will explore the analysis of such systems in detail. Our hypothesis is that the set of inflectional vectors maintains a rigid relative structure across these different patterns, but that they are rotated in various ways.

Nuer is a language with case marking on nouns for three cases (nom, gen, loc) and sg/pl, and several different inflectional classes of nouns.

As discussed in detail by Baerman (2012), the paradigms of number and case suffixes on nouns in Nuer vary among at least sixteen different classes, with similar but not quite identical patterns occurring among the classes. The following table, taken from Baerman (2012), which he adapted from Frank (1999), illustrates the complexity of variation among these classes.

(59) -

	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII
NOM SG	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅
GEN SG	∅	kä	kä	∅	∅	∅	kä	∅	kä	∅	kä	kä	∅
LOC SG	∅	kä	kä	∅	∅	kä	∅	∅	∅	kä	kä	kä	∅
NOM PL	∅	∅	ni	ni	∅	∅	ni	∅	∅	ni	∅	∅	∅
GEN PL	ni	ni	ni	ni	∅	ni	ni	ni	ni	ni	ni	∅	∅
LOC PL	ni	ni	ni	ni	∅	ni	ni	∅	ni	ni	∅	∅	ni
# of lexemes	61	52	45	23	11	10	9	8	5	3	2	2	2

	XIV	XV	XVI	XVII	XVIII	XIX	XX	XXI	XXII	XXIII	XXIV	XXV	
NOM SG	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	∅	
GEN SG	kä	kä	∅	ä	ä	kä	kä	∅	ä	∅	kä	kä	
LOC SG	kä	∅	kä	ä	ä	ä	ä	ä	kä	ä	kä	kä	
NOM PL	∅	∅	∅	ni	∅	ni	∅	∅	∅	ni	ni	∅	
GEN PL	∅	ni	∅	ni	ni	ni	ni	ni	ni	ni	∅	kä	
LOC PL	∅	∅	ni	ni	ni	ni	ni	ni	ni	ni	ni	ni	TOTAL:
# of lexemes	1	1	1	4	2	2	2	1	1	2	1	1	236

To simplify the analysis, we abstract away here from consideration of suffix *ä*, which occurs only among 13 out of 252 lexemes, and focus on the first sixteen classes in the table.

As Baerman remarks, the paradigms look deceptively simple, with only three suffixes occurring among the majority of classes.

> "On the face of it this is a very simple system. But consider how these suffixes are distributed in the paradigms of individual nouns, some examples of which are given in Table 3. With some lexemes the suffixes are restricted to a single morphosyntactic value; with others they are SYNCRETICthat is, they combine two or more distinct morphosyntactic values in a single form. For example, -k is used for the genitive singular of 'potato', but for the genitive and locative singular of 'bump'. While variation between syncretic and nonsyncretic distribution of morphological formatives is to be found in many languages, the sorts of patterns found in the Nuer paradigms are not ones that current models of morphology are well equipped to describe." (Baerman 2012, 468)

The challenge presented by Nuer is to account for the paradigmatic variation among classes through the simplest possible model of grammar. In the model we are proposing, there is in effect no syncretism: every suffix carries some value for every morphosyntactic feature by the very nature of the model. Which suffix is realized for a particular combination is determined by competition among suffixes with respect to their projection on a vector for that feature-value combination. What is referred to as 'blocking' in other models arises naturally in this model as a result of the competition.

How, then, does the grammar account for paradigmatic variation among classes? Our proposal, as stated above in §1, is that there is a configuration of vectors for the suffixes which remains constant across classes and which varies only by rotations that are applied to the vectors, with each rotation applying equally to each vector in the case of Nuer. These rotations preserve both the lengths of the vectors and the angles of separation between them. All the learner needs to derive a given class is the base configuration of vectors plus the rotation applied to that configuration that takes it to its position for that class.

4.2 Rotations: Deriving inflection classes with rotations

In the same manner as we chose initial weights for vectors for the German verb in (21), we can choose base weights for all the classes by adding up the number of times each suffix occurs for a given feature in the table in (59) and weighting the number by the number of lexemes that represent the class in which we are counting. This gives us the following counts if we include the classes that have at least 3 lexemes.

(60)

	sg	pl	nom	gen	loc
∅	460	177	374	127	126
ni	0	510	80	218	212
kä	234	0	0	119	114

There are advantages to applying L_2 as opposed to L_1 normalization. The former creates vectors of unit length for each suffix. Here are the results of L_2 normalization.

(61)

	sg	pl	nom	gen	loc
∅	.714	.275	.580	.197	.195
ni	0	.851	.133	.364	.354
kä	.817	0	0	.416	.398

These weights for suffix vectors result in the following activations, for which the maximum values for each combination result in the paradigm for Class 3.

(62)

	∅	ni	kä
nom.sg	**1.294**	0.133	0.817
gen.sg.	0.911	0.364	**1.233**
loc.sg.	0.909	0.354	**1.215**
nom.pl.	0.855	**0.984**	0.000
gen.pl.	0.472	**1.215**	0.416
loc.pl.	0.470	**1.205**	0.398

Class 3 only has the third highest number of lexemes, but we can think of the vectors in (61) as representing a central position in the space in which the vectors occur, inasmuch as they are derived from a weighted average of positions. We can therefore consider this set of vectors to be a set of base positions from which the other classes can be derived by rotations.

4.3 A learning algorithm for rotations

The following algorithm was able to produce weights for suffixes that give the correct paradigm for each of classes 1 through 16 by applying, for each class, the same rotation to each of the three suffixes, starting from the class 3 base configuration given above. Here are the main features of the algorithm.

- Approximate a transformation through all dimensions in the direction of the activations we want by breaking down each transformation into six 2D rotations.
- Each rotation moves the vectors in a direction determined by the ideal state we want for a particular morphosyntactic combination.
- Do this for each combination regardless of whether it already has the correct suffix with maximum activation, based on the current set of weights.[17]
- We can weight the amount that we want to affect each combination according to how far the intended winner is from the actual winner at a given point:

[17] The reason for this is that a rotation that is intended to affect just one feature combination will also have an effect on other combinations whose features are in the subspace in which we are rotating.

– If the intended winner is far behind the actual winner, make a more strongly weighted change in a direction that would increase the activa- tion of the intended winner for that morphosyntactic combination and decrease the activation for the wrong winner.
– If the intended winner is already winning, increase its activation for that morphosyntactic combination by a weaker amount.

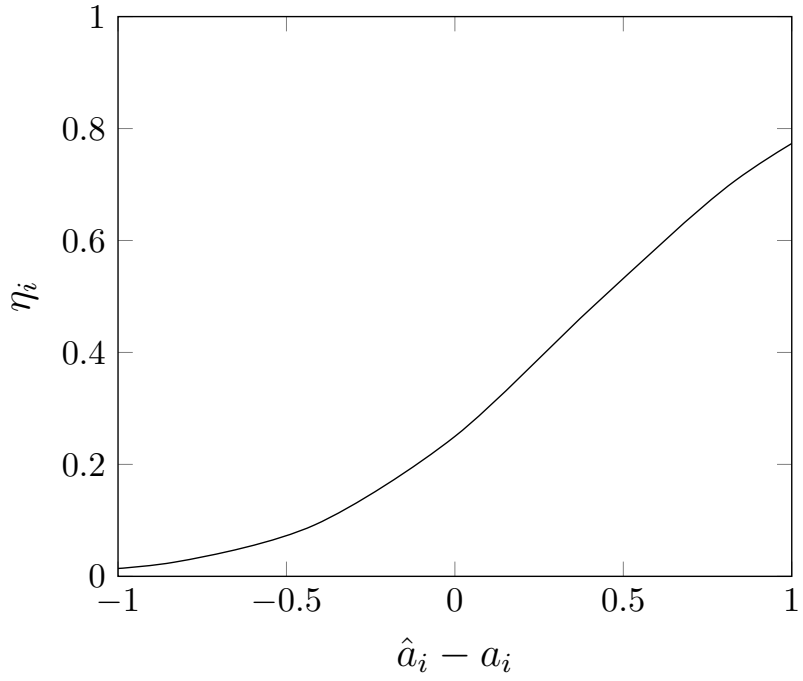
If the independent variable is the activation of the real winner minus the activation of the intended winner, a function that comes to mind that fits the pattern we want is a sigmoid function, which has a low value when the independent variable is less than 0 (the intended winner is already winning) and goes up sharply to a higher value when the independent variable is greater than 0 (the intended winner is losing.) Squaring the function seems to produce an even more satisfactory result.

Here, η_i is the factor by which we multiply our rotational increment on each sub-iteration, $\hat{a}_i$ is the activation of the current real winner and a_i that activation of the intended winner for morphosyntactic combination i.

(63) $\eta_i = \frac{1}{\left[1+e^{-2(\hat{a}_i - a_i)}\right]^2}$

Here is how a graph of the function looks.

A modified sigmoid function



(64)

On each iteration, there are six sub-iterations, each of which considers, in turn, one of the six morphosyntactic combinations. The algorithm chooses two features/dimensions in which to rotate the suffix vectors:

1. The feature for which the intended winner's weight for that feature exceeds that of its closest rival by the maximum amount. We are going to rotate away from this axis, since here is where the intended winner can most afford to lose some weight.

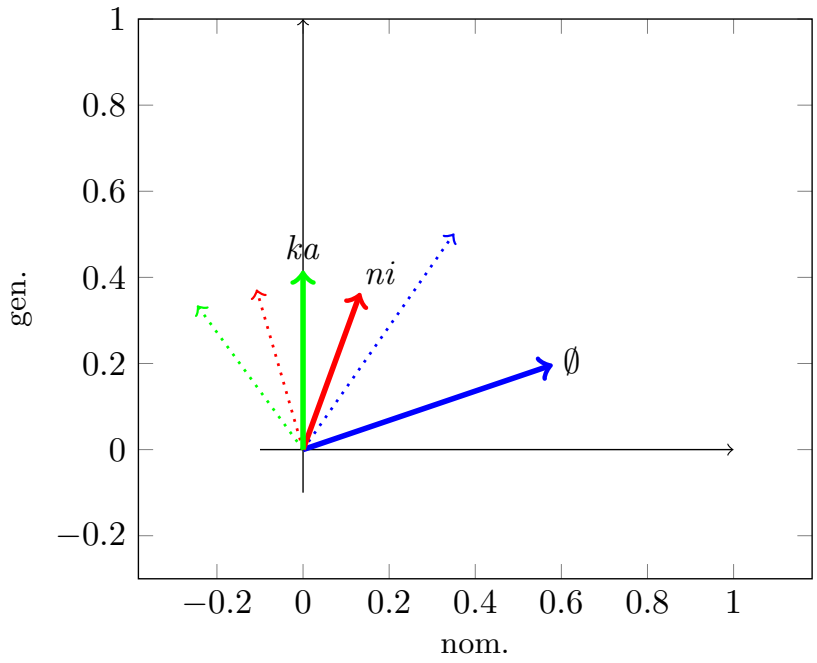
2. One of the two features, randomly chosen, that comprise the combination we are concerned with for this sub-iteration. We are going to rotate *towards* this axis.

The following 2D graph shows how this rotation works for one example. We are choosing a 2D space in which the x-axis is a feature in which the intended winner has greater value than its competitors and the y-axis in the dimension in which we want the intended winner to gain. Here it is for x=nom. and y=gen. with ∅ the intended winner and *kä* the false winner. Arrowed lines in solid colours indicate the original positions of vectors. Dotted lines in the same colour indicate the ending position of a vector

after rotation. We can see that a positive (i.e. counter-clockwise) rotation of the vectors will increase the projection of ∅ on the genitive axis as desired and decrease the projection of kä on the genitive axis. The fact that ∅ loses some of its projection on the nominative axis does not matter. The nominative axis was chosen to rotate away from since ∅ can afford to lose value there.

(65)

Graph of suffixes ∅ ni and kä in 2D: nom. gen.



4.4 Results of learning algorithm to derive 16 classes

In the following table, all classes were derived by rotation from the base position of vectors given above in (61) which corresponds to Class 3. The term 'distance' refers to the number of cells in the paradigm at which the class in question differs from Class 3. The term 'smallest margin' refers to the smallest difference in activation among all the cells in the paradigm for that class between a winning suffix and its closest competitor (i.e. projection onto the vector of a morphosyntactic feature combination). The algorithm was run 100 times for each class, with the results averaged.

(66)

Class	#lexemes	Dist. from base pos'n	Smallest margin	#iter
1	61	3	0.034	8.73
2	52	1	0.033	1.48
4	23	2	0.061	2.00
5	11	5	0.044	4.86
6	10	2	0.027	7.36
7	9	1	0.046	2.83
8	8	3	0.033	10.03
9	5	2	0.024	6.69
10	3	1	0.069	2.98
11	2	2	0.048	5.22
12	2	2	0.046	5.56
13	2	3	0.026	9.72
14	1	3	0.057	3.50
15	1	3	0.040	7.91
16	1	3	0.037	7.62

4.5 'Variable defaults' (Baerman 2012, 482)

Baerman's analysis of Nuer suffixes proposes to capture certain statistical preferences among the configurations that occur in Nuer through a set of 'variable defaults' that are subject to a set of implications. Some of the defaults he proposes are the following.

1. By default, genitive and locative singular are *kä*.
2. By default, genitive and locative plural are *ni*.
3. By default, nominative plural is ZERO.
4. If the nominative plural is *ni*, this entails *ni* in the other plural cases.
5. By default, genitive and locative are identical.

Most of the tendencies that Baerman observes will fall out naturally from our model without the need to posit default rules. Consider again the base values for suffix vectors we proposed in (61), repeated here as (67).

(67)

	sg	pl	nom	gen	loc
∅	.714	.275	.580	.197	.195
ni	0	.851	.133	.364	.354
kä	.817	0	0	.416	.398

1. Baerman's default rule 1 falls out from the fact that the projections of vector *kä* on gen.sg. and loc.sg. are .817 + .416 = 1.233 and .817 + .398 = 1.215 respectively, far greater than that of the other two suffixes. Consequently, the vectors would have to be rotated to a great extent in order to change this tendency.
2. Baerman's default rule 2 falls out in a similar fashion, where vector *ni* has projections of .851+.364 = 1.215 and .851 + .354 = 1.205 on vectors gen.pl. and loc.pl.
3. His rule 3 is not reflected as strongly in the values we have proposed, but neither is the presence of *ni* in nom.pl. as strongly attested among the classes.
4. The tendency towards rule 4 can be accounted for as follows. If nom.pl. is *ni*, as it is for our base class, then to maintain *ni* in the nominative plural we need to avoid rotations that reduce the strong weight of .852 in the plural for *ni*, since its weight in the nominative is less strong and cannot be depended on to maintain a strong projection for *ni* on the nom.pl. vector. Maintaining a strong weight in the plural for *ni* will tend to make it the winner for other combinations in the plural.
5. The tendency for genitive and locative to be identical is enforced by the values we have proposed for them among the three vectors, which are very close for all of them.

4.6 Deponent verbs: an example from Latin

We have shown that in our model, relations between different inflectional classes can be accounted for through rotations of a complete set of vectors for morphemes in the space of feature values. These kinds of rotations can also account straightforwardly for cases of deponency, which is a mismatch between morphosyntactic function and morphological form. A well-known example of this sort is the class of deponent verbs in Latin, which, in the present indicative active, take the set of suffixes that would normally occur for passive voice. The following data, taken from Stump (2016, 198) compare the paradigms of a non-deponent and a deponent first conjugation verb.

The active voice in the deponent verb is expressed with the passive voice suffixes and the passive voice does not occur with the deponent stem. The following matrices express the counts of each feature value for each affix in the paradigm of regular verb *parāre*. As we did for the Spanish verbs in §3.2, for simplicity of exposition we shall not treat theme vowels as separate affixes but consider suffixes such as *āmur* as a single morpheme.

		I PARĀRE 'prepare'
Active	1sg	*parō*
	2sg	*parās*
	3sg	*parat*
	1pl	*parāmus*
	2pl	*parātis*
	3pl	*parant*
Passive	1sg	*paror*
	2sg	*parāris*
	3sg	*parātur*
	1pl	*parāmur*
	2pl	*parāmini*
	3pl	*parantur*

		I CŌNĀRI 'try'
Active	1sg	*cōnor*
	2sg	*cōnāris*
	3sg	*cōnātur*
	1pl	*cōnāmur*
	2pl	*cōnāmini*
	3pl	*cōnantur*
Passive		(none)

Table 5: First declension Latin non-deponent and deponent verbs

(68)

	Active suffixes						Passive suffixes					
	o	ās	at	āmus	ātis	ant	or	āris	atur	āmur	āmini	antur
singular	1	1	1	0	0	0	1	1	1	0	0	0
plural	0	0	0	1	1	1	0	0	0	1	1	1
1st	1	0	0	1	0	0	1	0	0	1	0	0
(2nd	0	1	0	0	1	0	0	1	0	0	1	0
3rd	0	0	1	0	0	1	0	0	1	0	0	1
active	1	1	1	1	1	1	0	0	0	0	0	0
passive	0	0	0	0	0	0	1	1	1	1	1	1

If we normalize the columns, we have the following values, expressed algebraically:

(69)

	Active suffixes						Passive suffixes					
	o	ās	at	āmus	ātis	ant	or	āris	atur	āmur	āmini	antur
singular	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	0	0	0	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	0	0	0
plural	0	0	0	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	)	0	0	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$
1st	$\frac{1}{\sqrt{3}}$	0	0	$\frac{1}{\sqrt{3}}$	0	0	$\frac{1}{\sqrt{3}}$	0	0	$\frac{1}{\sqrt{3}}$	0	0
2nd	0	$\frac{1}{\sqrt{3}}$	0	0	$\frac{1}{\sqrt{3}}$	0	0	$\frac{1}{\sqrt{3}}$	0	0	$\frac{1}{\sqrt{3}}$	0
3rd	0	0	$\frac{1}{\sqrt{3}}$	0	0	$\frac{1}{\sqrt{3}}$	0	0	$\frac{1}{\sqrt{3}}$	0	0	$\frac{1}{\sqrt{3}}$
active	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	0	0	0	0	0	0
passive	0	0	0	0	0	0	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$

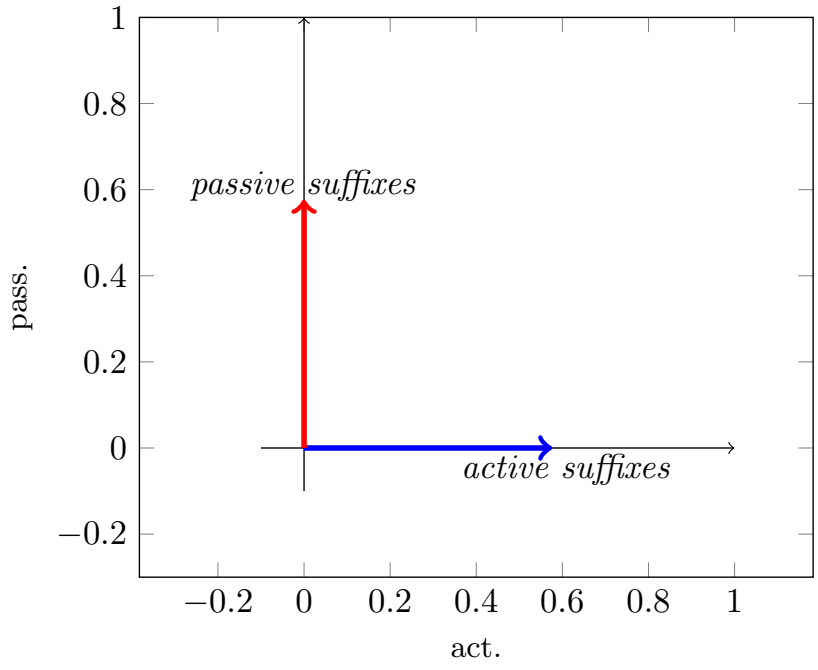
. . . and expressed numerically:

(70)

	Active suffixes						Passive suffixes					
	o	ās	at	āmus	ātis	ant	or	āris	atur	āmur	āmini	antur
singular	0.577	0.577	0.577	0	0	0	0.577	0.577	0.577	0	0	0
plural	0	0	0	0.577	0.577	0.577	0	0	0	0.577	0.577	0.577
1st	0.577	0	0	0.577	0	0	0.577	0	0	0.577	0	0
2nd	0	0.577	0	0	0.577	0	0	0.577	0	0	0.577	0
3rd	0	0	0.577	0	0	0.577	0	0	0.577	0	0	0.577
active	0.577	0.577	0.577	0.577	0.577	0.577	0	0	0	0	0	0
passive	0	0	0	0	0	0	0.577	0.577	0.577	0.577	0.577	0.577

We can see that there is an exact correspondence between the feature values of the active affixes and the passive affixes except for the active and passive values themselves. If we look at the 2D subspace of the active and passive feature values, all the active affixes will be in one position and all the passive affixes in another, as shown in the following graph.

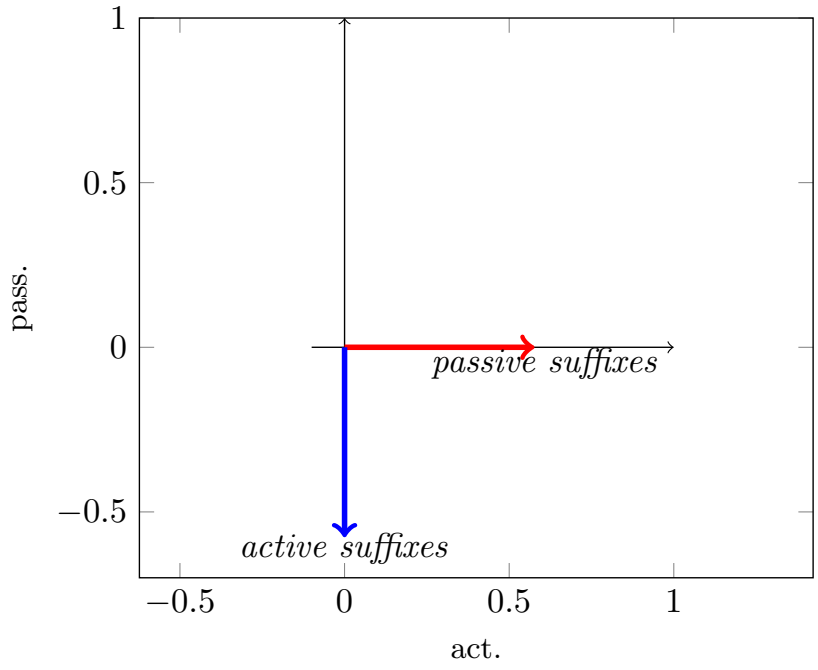
Graph of active and passive suffixes in 2D: activ. passive.



(71)

If we apply a rotation of a three-quarter turn counter-clockwise to all the affixes, just in that 2D subspace, we will end up with the following positions for the two sets of vectors.

Graph of active and passive suffixes in 2D: activ. passive.



(72)

The affixes that normally occur in the passive paradigm now have the exact values that the active affixes had and the active suffixes have moved to negative territory. This will result in the passive suffixes being chosen for feature-value combinations involving active voice – exactly what we see for deponent verbs. This rotation in the 2D subspace (active, passive) can be represented as an identity matrix that has the values in the active and passive dimensions (in this case 6th and 7th rows and columns) changed to the following submatrix where θ is the angle of counter-clockwise rotation:

(73)

$$\begin{array}{c} \\ \textit{active} \\ \textit{passive} \end{array} \begin{array}{c} \begin{array}{cc} \textit{active} & \textit{passive} \end{array} \\ \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \end{array}$$

In this case, the counter-clockwise rotation of a three-quarter turn is 270^o or $\frac{3\pi}{2}$ radians, whose sine and cosine are -1 and 0 respectively. This gives us the following rotation matrix, which, applied to the feature-value set for a regular verb will result in the set for a deponent verb. Cells whose values depart from those in an identity matrix are coloured blue.

(74) $$RotationMatrix = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

Here is how the matrix of feature values looks after applying this rotation. (Numbers are rounded to two decimal places to save space.)

(75)

	Active suffixes						Passive suffixes					
	o	ās	at	āmus	ātis	ant	or	āris	atur	āmur	āmini	antur
singular	0.58	0.58	0.58	0	0	0	0.58	0.58	0.58	0	0	0
plural	0	0	0	0.58	0.58	0.58	0	0	0	0.58	0.58	0.58
1st	0.58	0	0	0.58	0	0	0.58	0	0	0.58	0	0
2nd	0	0.58	0	0	0.58	0	0	0.58	0	0	0.58	0
3rd	0	0	0.58	0	0	0.58	0	0	0.58	0	0	0.58
active	0	0	0	0	0	0	0.58	0.58	0.58	0.58	0.58	0.58
passive	−0.58	−0.58	−0.58	−0.58	−0.58	−0.58	0	0	0	0	0	0

The feature values for the passive suffixes (right half of the matrix) are now exactly what they are for the active suffixes without the rotation.

5 Conclusions

This paper illustrates some of the initial results that arise from treating the problems of inflectional morphology from a geometrical point of view. From our perspective, the advantages of a geometrical perspective are three in number:

1. Despite the initial novelty of thinking of morphemes as vectors in a space of dimensionality greater than 3, it allows for a visually intuitive way of seeing how the structural information of morphemes interact with one another—how the inflectional information of a stem interacts with the information of a neighboring affix, for example. This kind of interaction includes a natural geometric account of why more specific morphemes typically dominate over more general morphemes: it is because they are closer to the target.

2. There is a large set of learning algorithms in machine learning that are easily applicable to models grounded in geometry.

3. The analysis is distinctly less derivational than what is found in analyses in some other approaches. We certainly recognize the importance of including several distinct representations in the analysis of a given word or utterance, and in that sense we are perfectly comfortable with the notion of a derivation in the abstract. But derivational accounts run certain risks, in our opinion. It has not been our intent in this paper to contrast our account with others, but we have tried to develop an account in which we avoid two things that are natural in a derivational context: (1) the use of rules that have been called *impoverishment* rules or feature-deletion, and (2) a style of explanation that employs a sort of abstract topography which aims to offer a linguistic explanation. The first we avoid because we are not sure that such rules are formally, or theoretically, coherent[18] The second is a style of explanation that we are uncomfortable with, which takes pedagogical metaphors as if they were meant in some sense literally, such as offering an explanation of a phenomenon by the saying that one computation occurs *here*, and another occurs *there* (one in the lexicon, say, and one in the syntax) and that it is this imaginary distance that provides an explanation of their ignorance of each other.

The validity of this approach will depend on whether it can be extended to the range of phenomena known to morphologists, and we invite our reader to join in that exploration.

6 Some remarks about learnability

6.0.1 Falsifiability and learnability

Karl Popper is often associated with the view that an important goal for philosophy is to provide a means for determining which human enterprises are sciences and which are not, and the view that a scientific

[18] At risk of over-simplifying, we take it that when we develop a theory, we employ objects, relations, and functions of various sorts. Features and feature values are distinct sorts of entities, and features *may* be understood as functions from objects to a set of feature values. Functions are not the sort of entity that delete (or are deleted) by virtue of their nature of mapping from one domain to another co-domain. It seems to us that care was employed in developing the theory of autosegmental phonology so as not to risk inadvertent theoretical incoherence.

theory must provide explicit means of proving itself wrong. Popper was concerned that such fields as Freudian psychology and Marxist economics were not scientific, and the justification for this belief lay in the fact that no one was able to specify observations that would prove either theory wrong. Regardless of what one thinks of this solution to what Popper called the problem of demarcation, it is a simple error—a misunderstanding—to think that something like it can be used as a measure for declaring one theory *more* scientific or *less* scientific than another. Popper's solution was not intended (and should not be understood as) an apriori evaluation of a scientific theory's desirability.

6.0.2 Ignorance of non-existence

The second reason that we should avoid preferring theories that exclude grammars over those that do not is that as a discipline, we do not have measures in place for evaluating claims about the existence or non-existence of particular phenomena across the range of human languages. Broad claims are not infrequently made by linguists motivated to rule out linguistic phenomena, and the only means to check the validity of their claims is the hope (if that is the proper word) that someone who reads their work and knows a counterexample is motivated enough to inform them of it. Many linguists operate by the principle that a linguist has the freedom to make a claim (as if being granted permission by the late Sir Karl Popper) by virtue of the fact that someone else might do the work to show them wrong. This is not a reasonable way to run a science.

6.0.3 Expansionary phase

The third reason is tightly connected the second: linguistics as a field is still in an expansionary stage, in the sense that any linguist who spends a year or two studying a language will inevitably discover a new phenomenon that is new and interesting. A student who studies a language and only finds phenomena that match perfectly what has been described in the literature that they are exposed to is not a linguist; we are certain that they have not looked hard enough.

Closely related to that, too, is the fact that there have been no important publications in the field to date whose primary contribution is the elimination of a certain aspect of formal grammar. All important works either expand our universe of known linguistic phenomena, or they provide simpler and more insightful accounts of what we already were aware of.

6.0.4 Learning is not random selection

It was perhaps once reasonable to think that the discovery of limitations on the class of possible human languages would shed light on how language is learned, but that time is long gone, and has been gone since the beginning of machine learning in the 1980s. We now know a great deal about learning, especially in the computational context, and we know about many ways in which structure can be inferred from data. [finish]

7 Appendix: summary of important variables mentioned

We summarize our use of different spaces so far in the following tables. Note that *number of entries* is in a linguistic sense the number of degrees of freedom in the system; it is the number of distinct entries in the entire description of the paradigm.

Parameter	Symbol	Illustrated
Number of morphemes	NumMorph	# columns in $\mathcal{B}$
Number of feature values in paradigm space	NumFeaVal	# rows in $\mathcal{B}$ or columns in Φ.
Number of inflectional features	InflDimen	
Number of positions in paradigm	NumParaPos	# rows in Φ

Table 6: Important parameters

	Number of dimensions	Number of feature values	number of entries
PS: *paradigm space*	number of inflectional features (InflDimen)	sum of possible values of inflectional features (NumFeaVal)	product of possible values
English weak verbs	3	7	12 (each a morpheme).
VS: *feature value space*	number of feature values in feature space NumFeaVal	2 in each dimension (0,1) (trivially)	number of morphemes × number of feature-values
English weak verbs	7	2 in each dimension for paradigm points	3 × 7 (each a real number). NumMorph × NumFeaVal.

Table 7: Paradigm space versus Feature-value space

References

Scott Aaronson. *Quantum Computing Since Democritus*. Cambridge University Press, 2013.

Matthew Baerman. Paradigmatic chaos in Nuer. *Language*, 88(3):467–494, 2012.

Greville G. Corbett and Andrew Fraser. Network Morphology: A DATR account of Russian nominal inflection. *Journal of Linguistics*, 29(1):113–142, 1993.

Roger Evans and Gerald Gazdar. An introduction to the Sussex Prolog DATR system. (Cognitive Science Research Report CSRP 139). In Roger Evans and Gerald Gazdar, editors, *The DATR Papers*, pages 63–71. University of Sussex, 1989.

Wright Jay Frank. Nuer noun morphology. Master's thesis, State University of New York, Buffalo, 1999.

Norman M. Fraser and Richard Hudson. Inheritance in Word Grammar. In *Computational Linguistics*, volume 18, pages 133–158, 1992.

John Goldsmith. Grammar within a neural net. In *The Reality of Linguistic Rules*, pages 95–113. John Benjamins, 1994.

John Goldsmith and Eric Rosen. Learning morphosyntactic categories and features for inflectional paradigms. Submitted to the 2017 annual meeting of the Association for Computational Linguistics.

Graham Russell, Afzal Ballim, John Carroll, and Susan Warwick-Armstrong. A practical approach to multiple-default inheritance for unification-based lexicons. In *Computational Linguistics*, volume 18, pages 311–337, 1992.

Edward Stankiewicz. *Declension and gradation of Russian substantives in contemporary standard Russian*. Mouton, The Hague, 1968.

Gregory Stump. *Inflectional Paradigms, Content and Form at the Syntax-Morphology Interface*. Cambridge University Press, Cambridge, 2016.

Boris O. Unbegaun. *Russian Grammar*. Oxford University Press, Oxford, 1957.

V. V. Vinogradov, E. S. Istrina, and S. G. Barxudarov, editors. *Grammatika russkogo jazyka, vol. I: Fonetika i morfologija*. ANSSR, Moscow, 1952.