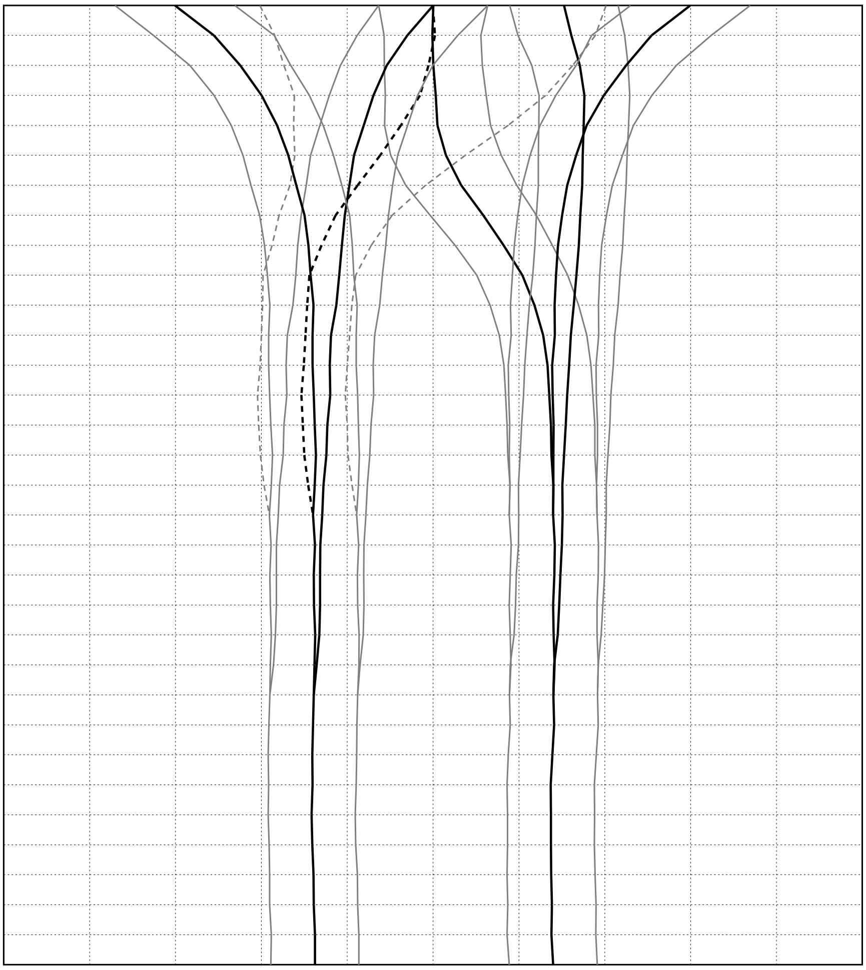
The evolution of auditory dispersion in symmetric neural nets



Thesis, research master Linguistics, University of Amsterdam. Student: Klaas Seinhorst; supervisor: prof. dr. Paul Boersma; second reader: dr. Silke Hamann. August 10, 2012.

[…]

The clouds exaggerate and pile
into heights of mile on mile.
In the breathing o' the universe
they drift asunder and disperse.

(Paul Goodman, set to music by Ned Rorem)

Index

1 Introduction

The title of my thesis will undoubtedly ring a bell with some readers. This is no coincidence: it has the same goals as an article by Paul Boersma and Silke Hamann, 'The evolution of auditory dispersion in bidirectional constraint grammars', which appeared in *Phonology* in 2008. These goals are: (i) formalizing auditory dispersion as an interaction of a prototype effect and an articulatory effect, and (ii) simulating the evolution of auditory dispersion over a number of generations for different types of initial distributions. Both this thesis and the article by Boersma and Hamann make use of the same theoretical model, namely Boersma's BiPhon model: the difference lies in the computational framework used. Whereas Boersma & Hamann used bidirectional constraint grammars, this thesis relies on symmetric artificial neural networks or *neural nets*, aiming to further the coupling of this framework with the BiPhon model (for instance, semantic dispersion effects were simulated by Boersma 2009a). Also, this thesis is intended as an exercise for myself in writing scripts and running computer simulations of language evolution.

This thesis is made up of two major sections, namely Part I: Theory and Part II: Application. Their internal structures are as follows: Part I contains an exposition of the phenomenon of auditory dispersion (§2), an overview of relevant literature (§3) and an introduction to neural nets (§4); Part II sets out the architecture of the network used for the computer simulations and describes the scripts (§5), presents the results of the simulations (§6) and discusses the issue of learnability, including a rudimentary strategy to quantify it (§7). After Part II follow a conclusion and discussion (§8), and an Appendix in which the scripts used for the simulations are compiled.

PART I: THEORY

2 Auditory dispersion

It has often been noted that languages structure their phoneme inventories in a way that maximizes contrast between categories (e.g. Passy 1890; Von der Gabelentz 1901; De Groot 1931; Martinet 1960). Such inventories are optimally *dispersed*. The notion of contrast has been explained as pertaining to distinctive phonological features and their (usually binary) values, e.g. [–back] or [+high], or to acoustic-phonetic cues, e.g. the Voice Onset Time (VOT) in plosives or the first formant (F_1) in high vowels. Both perspectives are illustrated in Figure 1, depicting the minimal three-vowel inventory /i a u/:

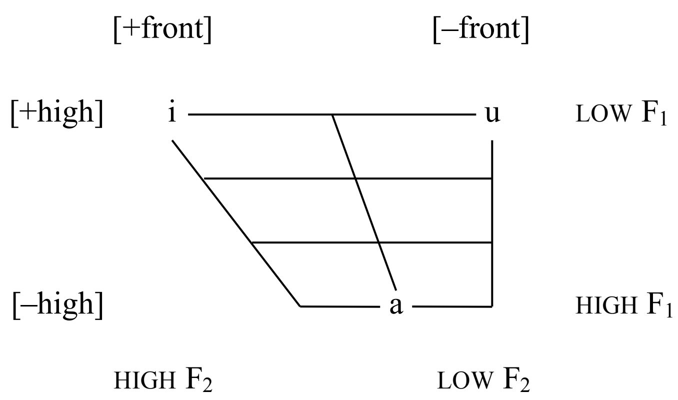


Figure 1. *Dispersion in the vowel space: phonological and phonetic explanations. Binary features in [square brackets], auditory cues in* SMALL CAPITALS.

This inventory is optimally dispersed with respect to the relevant phonological features: /a/ is [–high]; /i/ and /u/ are both [+high], but /i/ is [+front] and /u/ is [–front]. Similarly, this inventory is optimally dispersed with respect to the frequency values of its first two formants: /a/ has a high F_1; /i/ and /u/ have a low F_1, but /i/ has a high F_2 and /u/ has a low F_2.

The preference for optimally dispersed phoneme inventories is generally explained by a basic functional principle: if categories are sufficiently distinct, confusion in the listener is minimized, which aids successful communication.

I assume that auditory cues – the raw data by which the language learner is surrounded (and to which she is in fact already exposed prenatally) – play a crucial role in dispersion, and use the term *auditory dispersion*. Feature-based explanations often assume one of the values of the feature to be marked (Jakobson 1941; Prince & Smolensky 1993/2004), which entails that a language has either both the unmarked and marked segment, or only the unmarked, but never only the marked. However, this assumption is false, as some languages do have only marked segments: Russian, for instance, has (marked) [n^j] and [n$^\gamma$] but not (unmarked) [n]. This can only be due to the auditory contrast between [n^j] and [n$^\gamma$] (Padgett 2001, 2003).

Another issue with features is the fact that they are covert: this means that feature-based explanations have the burden of explaining their origins. It is generally, and often tacitly, assumed that they are universal and innate, but this assumption is troublesome. It has been suggested that a restricted set of innate features or properties, as proposed by adherents of innatism, cannot account for typological observations (e.g. Boersma 1998, who argues that if sonority scales were innate, they would be identical for different purposes, but they vary according to the functional demands of these purposes (§19.2.2); Mielke 2004, who found that allegedly universal features have little success in explaining the structure of phonological inventories of a sample of languages), and it is my impression that the appearance of innateness as an explanatory device is motivated more by the need for a *deus ex machina* than by empirical evidence. Since language systems are the products of a highly complex interplay between evolutionary processes (both biological and cultural), human biases (articulatory, sensory and cognitive), functional pressure, social factors and mere chance, which have been interacting for an immense stretch of time, they cannot be expected to be perspicuous. Laboratory experiments and computer simulations investigating language evolution without the assumption of domain-specific mechanisms or features are fairly recent, but have already yielded promising results: e.g. population structure bears an influence on the complexity of a phonological inventory (observation by a.o. Lupyan & Dale 2010; simulations by Lopopolo 2011); properties of natural languages emerge in a process of iterated learning in the laboratory (distinctive features: Verhoef & De Boer 2011a; compositionality: Kirby, Cornish & Smith 2008; systematicity: Theisen, Oberlander & Kirby 2010); age-dependent differences in the rate of language acquisition influence language change and stability (simulations by Verhoef & De Boer 2011b). Computer simulations of the acquisition of the phonology-phonetics interface have also achieved realistic results (e.g. Guenther & Gjaja 1996; Boersma, Escudero & Hayes 2003; Wanrooij 2009; Boersma, Benders & Seinhorst fc.).

Lastly, innate features would probably only obstruct the learning process: they force the child to take an additional step – learning which features connect to which properties of the input data, and which do not connect to anything; also, identical feature values relate to different auditory events cross-linguistically, and this language-specific mapping would have to be learned.

The hypothesis that auditory contrast shapes phonological systems makes the prediction that if a language has two categories on a given perceptual continuum, they are at its edges, as in Figure 2:

Figure 2. *Maximal perceptual contrast: not attested.*

This prediction, however, does not hold true. The distribution depicted in Figure 2 is (to my knowledge) not found in languages. This is because peripheral values of an auditory continuum are probably harder to articulate than central values (Boersma & Hamann 2008 assume a parabolic articulatory effort curve). Humans, however, prefer the most efficient way to achieve their goals (Zipf 1935, 1949), i.e. the way that yields maximal results at minimal physical cost: this means that the production of peripheral auditory values is disfavoured if possible. Rather, the categories in an inventory are inclined towards the centre of the continuum, while maintaining a sufficient distance between them. An example of such a division is depicted in Figure 3:

Figure 3. *Maximal perceptual contrast with minimal articulatory effort: attested.*

The degree to which categories can be inclined towards the centre of the continuum is dependent on the total number of categories: $n + 1$ categories occupy a larger part of the auditory continuum than n categories. For instance, languages tend to divide the auditory continuum of the first formant (F_1) in front vowels in one of the following ways (very few languages have more than four distinctive vowel heights):

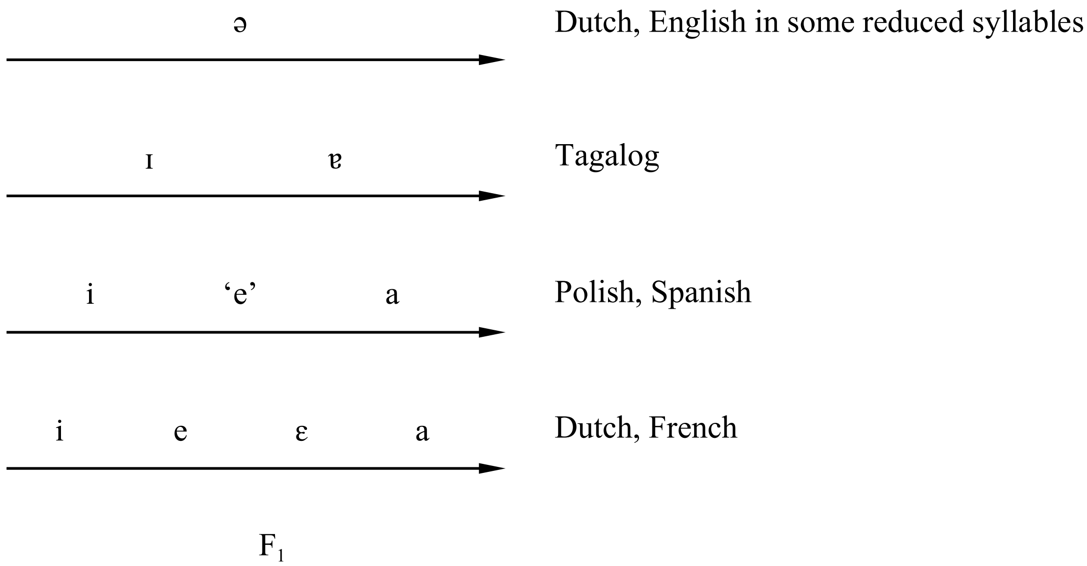


Figure 4. *Distributions of vowel categories along the F_1 continuum.*[1]

The observation that larger inventories occupy a larger part of the phonetic space is also valid in two-dimensional cases, e.g. the vowel space, slightly simplified here to a triangle with blunt corners:

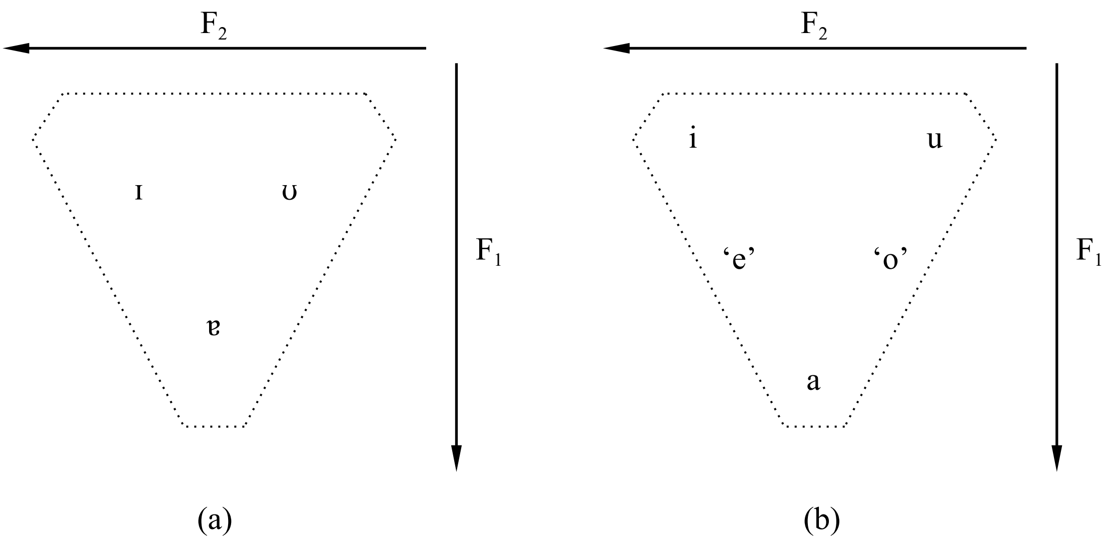


Figure 5. *Typical distributions of vowel categories across the vowel space (Schwartz et al. 1997): three-vowel languages (a) and five-vowel languages (b).*

[1] Single quotes ‘’ indicate that no exact phonetic transcription is available: for instance, ‘e’ indicates an unrounded front vowel with a central height, i.e. intermediate between e and ɛ.

In languages with three vowels, such as Tagalog, the categories occupy less peripheral locations in the auditory space than they do in languages with five vowels, such as Spanish (Boersma 1998: 216): this is because the former have fewer categories between which perceptual contrast must be maintained, which gives room to a larger influence of the propensity towards less effortful articulations.

Figures 4 and 5 show that phonological categories are usually evenly spaced and distributed around the central value of the auditory space, the realization that requires least articulatory effort. Hence, optimally dispersed systems strike an optimal balance between perceptual contrast and articulatory ease. In systems that are not optimally dispersed, sound change is predicted to take place, until optimal dispersion is achieved. This happened for instance in Polish, that used to have a set of voiceless sibilants [ʃ sʲ s] before the 15th century (Stieber 1952, 1973).[2] The spectral centres of gravity of [sʲ] and [s] differ only by a small amount; this confusing opposition was resolved by a contrast-enhancing shift of the spectral mean of [sʲ], lowering it to [ɕ]. However, another confusing opposition resulted from this shift, viz. [ʃ] vs. [ɕ]: this was resolved by a second sound shift in which [ʃ] retroflexed to [ʂ], approximately in the 16th century (Padgett & Żygis 2003). The resulting optimally dispersed system [ʂ ɕ s̪] is still found in contemporary Polish.

The structure of phoneme inventories may also be influenced by other than auditory factors. For instance, if a language has three high vowels, it is likely to favour the set [i ɨ u], which is optimally dispersed with respect to the second formant (F_2). The set [i y u], however, is also relatively common cross-linguistically: the former set (without [y]) is found in 80 out of 628 languages (12.7%) in Mielke's (2008) segment inventory P-base, the latter (without [ɨ]) in 37 (5.9%). [i y u] is not optimally dispersed with respect to F_2, but it may have a higher rate of occurrence than expected on the basis of its auditory characteristics because of the clearly visible lip rounding in [y]. This salient and reliable perceptual property of [y] may work to its advantage in phoneme inventories.

[2] [ʃʲ sʲ s] according to Żygis & Padgett (2007).

3 Formalizations of auditory dispersion

This chapter discusses a number of sources that have formalized and simulated dispersion in terms of auditory contrast. Most of these explanations include articulatory forces as well. Some accounts based on featural contrast are, for instance, Jakobson & Halle (1956); Clements (2003, 2005); Hall (2011). A theory that is concerned with the interaction of auditory and articulatory factors, but does not include auditory contrast, is *Quantal Theory* (Stevens 1972, 1989; Stevens & Keyser 2010). This theory argues that languages favour phonemes that are robust against articulatory imprecision, i.e. phonemes whose auditory correlates remain more or less the same even with fairly inaccurately executed articulatory gestures.

Most formalizations and simulations of auditory dispersion have focused on vowel systems (e.g. Liljencrants & Lindblom 1972; Lindblom 1986; Ten Bosch 1991; De Boer 1999, 2000, 2001; Boersma 2007; Van Leussen 2008). A smaller number treats consonant inventories or consonant classes (inventories: van Leussen 2009; classes: Boersma & Hamann 2008; Van Leussen & Vondenhoff 2008). Modelling entire consonant inventories has turned out to pose more problems than modelling vowel inventories, perhaps because the interaction of auditory cues is more complex in consonants than in vowels. Ohala (1980: 185) argues that maximum auditory contrast in consonant inventories would predict an improbable set of phonemes, for instance /ɗ k’ ts ɬ m r ɹ/, and asks whether consonant inventories may have been shaped by other factors than vowel inventories; Lindblom (1986) confidently answers ‘no’ to this question, but does not attempt a formalization, and neither do Lindblom & Maddieson (1988). While the categories in the set proposed by Ohala are indeed perceptually highly distinct, the cost of this contrast in terms of articulatory effort is disproportionate: hence I find it strange that Lindblom (1986: 41) calls Ohala’s argument “convincing”, considering his awareness of the relevance of articulatory factors (Liljencrants & Lindblom 1972: 856).

The endeavour of formalizing the interaction of auditory contrast and articulatory ease in consonant inventories was taken up by Van Leussen (2009), who found some promising results, but had rather limited resources at his disposal. A lot of further research is required in this area.

The sources discussed in this paragraph are divided into three categories: (i) research that does not make use of Optimality Theory (hereafter OT), treated in §3.1; (ii) research that does rely on OT, treated in §3.2; and (iii) a separate section (§3.3) is devoted to research carried out within the BiPhon model, most notably Boersma & Hamann (2008), the article on which this thesis builds.

3.1 Formalizations without OT

To my knowledge, the earliest attempt to simulate auditory dispersion is Liljencrants & Lindblom (1972). These authors draw an analogy between vowel categories and particles or magnets with an equal electrical charge: a stable distribution of a number of particles or magnets in a preset space is one in which the energy in the space is minimal. In general, this will correspond to a distribution in which the mutual distances are maximal. In the initial distribution in Liljencrants & Lindblom's simulations, the categories lie on the circumference of a circle that is located in the centre of the vowel space; for each category, an algorithm calculates in which direction the categories should move in order to minimize the energy in the space. This strategy is analogous to the force of maximization of perceptual contrast. For small numbers of vowels (ca. 3–6), the simulation results show considerable agreement with attested vowel inventories.

The used algorithm is not devoid of teleology, as the categories are explicitly repelled from one another. A second disadvantage is that Liljencrants & Lindblom do not take the force of articulatory ease into account, so that their algorithm predicts exaggerated vowel systems (e.g. [i a u] for three-vowel inventories, which tend to be the reduced inventory [ɪ ɐ ʊ], as in Figure 5a, p. 8). The authors acknowledge this shortcoming, but argue not having been able to avoid it, as they did not see a way to quantify articulatory effort.

Hall (2007) discovered another detriment of Liljencrants & Lindblom's approach: the final distribution of the categories seems to be highly dependent on their initial distribution. If the categories are not yet aligned with their final positions, the algorithm yields uncommon or even unattested inventories, such as the (more or less) vertically distributed set [ɨ ə a], the (more or less) horizontally distributed set [ɛ ə ɔ] or the skewed triangular distribution [ʉ æ ɑ] (Hall 2007: 145, 146). As far as I'm aware, it has not been tried to find out what Liljencrants & Lindblom's algorithm predicts if the vowel categories start at random locations in the vowel space, instead of on a circle in the centre. However, Hall's findings are discouraging in this respect: if random locations on the circumference of a circle in the centre of the space do not yield realistic results, random locations in the entire space are not likely to do so either. Also, if Liljencrants & Lindblom's approach were extended to a unidimensional space, representing a single auditory continuum, it would only make accurate predictions about one-category systems if this category starts in the middle: after all, in other cases it will not be pushed to the centre by other categories or articulatory forces.

Lindblom (1986) builds on his previous work with Liljencrants by improving the distance metric and incorporating listener-oriented and social factors into the model, but this improves the results only slightly. Ten Bosch (1991) incorporates the notion of articulatory effort into his model by means of a vocal tract model and an effort function to calculate the effort for any configuration of the model. He finds that repeatedly maximizing the distance between the two closest categories provides better results than Liljencrants & Lindblom's strategy of finding the energy minimum over the entire space.

The sources discussed above model dispersion within a single generation. In contrast, it is described and/or modelled as an emergent, evolutionary phenomenon by a.o. De Boer (1999, 2000, 2001), Blevins (2004) and Wedel (2006).

De Boer (1999, 2000, 2001) investigates the hypothesis that vowel inventories are shaped by self-organization, a phenomenon in which higher-level organized behaviour emerges as an unintended result of interactions of lower-level entities. He simulates the evolution of vowel systems in a population of interacting agents who negotiate on a vowel system that enables successful communication. In such systems, the vowels are required to be robust to noise-induced fluctuations in formant frequencies. The model does not use teleological devices to achieve dispersion: in fact, vowels that approach each other closely are merged (although this has the undesirable result that push chains cannot be explained). Some unrealistic issues remain, such as the random addition of new vowels to the inventory, but De Boer's results show that self-organization is likely to be an important factor in the emergence of optimally dispersed vowel systems.

Blevins (2004) and Wedel (2006) work with exemplar models, which posit that a listener stores all perceived tokens of any category in her episodic memory. A phonological category can thus be imagined as a cloud of phonetically detailed tokens. Any exemplar can be re-used in production: the probability that a token is selected depends on how recently it was stored, with recent tokens being more likely to be selected. In production, a small amount of noise is added to the auditory cues of the exemplar, and through a production-perception loop this produced token is stored in the speaker's memory as well. In principle, this would entail that categories can come to overlap and eventually blend, but Wedel (2006) proposes that ambiguous tokens – i.e. tokens that could belong to multiple categories – contribute less, or not at all, to future pronunciations. This way, the lexicon keeps categories distinct. Wedel bases this argument on psycholinguistic evidence that listeners recognize words with many lexical neighbours less quickly than words with few or none lexical neighbours (Luce & Pisoni 1998): however, Luce & Pisoni's findings do not depend on specific auditory cues.

3.2 Formalizations with OT

The advent of OT (Prince & Smolensky 1993/2004) brought a large number of accounts of auditory dispersion in this framework, e.g. Flemming (1995/2002); Kirchner (1998); Padgett (2003); Sanders (2003); Boersma & Hamann (2008); Hall (2011). Most of these accounts again model dispersion as a synchronic phenomenon by means of teleological constraints, whose sole purpose is to maintain a certain perceptual distance between categories, e.g. Kirchner's DISP, Flemming's MINDIST or Padgett's SPACE. The former two require a certain minimal auditory distance to exist between categories, the latter requires only one category to occupy a certain part of the phonetic space.

A disadvantage inherent to the use of teleological devices – whether they are dispersion constraints, repelling algorithms, or a lexical bias against storing confusing tokens as exemplars (if we accept Wedel's argument) – is that the formalizations using them cannot explain sound changes in which categories approach each other, such as mergers or push chains. These would require additional, counteracting constraints and different constraint rankings, whose relation to the dispersion grammar cannot be made clear. De Boer's model can handle mergers (and is non-teleological), but it cannot explain push chains.

3.3 The BiPhon model

A general disadvantage of most models discussed so far is that they are crafted for the specific purpose of simulating auditory dispersion, and would either require serious extension or adjustments to account for other basic phonological phenomena, or would fail to do so altogether. A model intended to describe all of a language user's phonological and phonetic knowledge is Boersma's BiPhon model (for an overview, see Boersma 2011), already briefly mentioned in the introduction. In their article in *Phonology*, Boersma and Hamann (hereafter, both this article and its authors will be abbreviated to B&H) embed their formalization of auditory dispersion in this model, and show that optimal auditory dispersion emerges innocently, i.e. non-teleologically.

The BiPhon model assumes separate but closely intertwined phonological and phonetic representations. The layout of the model is shown in Figure 6:

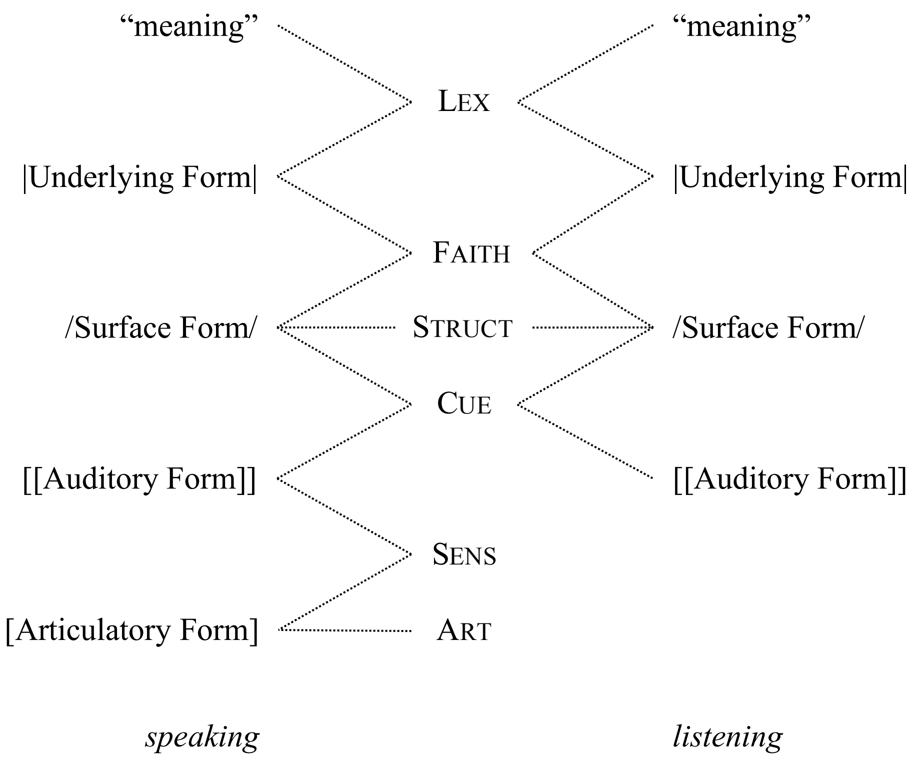


Figure 6. *The BiPhon model.*

"meaning" is a semantic representation; the Underlying Form (UF) and Surface Form (SF) are discrete phonological representations, familiar from Prince & Smolensky (1993/2004) and McCarthy & Prince (1995); the Auditory Form (AudF) and Articulatory Form (ArtF) are continuous phonetic representations. The Underlying Form is a lexical morphemic form; the Surface Form is structured in terms of feet, syllables, features, segments etc.; the Auditory Form (AudF) is a representation of auditory events such as pitches, silences, noise etc., and their ordering in time; the Articulatory Form (ArtF) is a gestural plan to realize the auditory events specified in the AudF (and hence only applies to the speaker). In OT-based formalizations, the levels are linked or evaluated by families of OT constraints: lexical constraints (LEX in Figure 6: Boersma 2001; Apoussidou 2007) relate morphemes to meanings; faithfulness constraints (McCarthy & Prince 1995) evaluate the relation between UF and SF; structural constraints (STRUCT: Prince & Smolensky 1993/2004) evaluate the phonotactic well-formedness of the SF; cue constraints (CUE: Escudero & Boersma 2003; Boersma 2009b) relate phonological surface structures to auditory events and vice versa; sensorimotor constraints implement the muscle movements needed to produce auditory cues (SENS; Boersma 2006); articulatory constraints (ART: Kirchner 1998; Boersma 1998) militate against articulatory effort, with central articulations being least effortful (Boersma 1998; Boersma & Hamann 2008).

All constraints apply bidirectionally, i.e. both in production and comprehension, with the same rankings in both directions. From the relevant input (i.e. the meaning in production, and the AudF in comprehension), the output (i.e. the ArtF in production, and the meaning in comprehension) and intermediate representations are computed in parallel.

In the BiPhon model, optimal auditory dispersion emerges as a side effect of the interaction of two functional forces in speech production: a *prototype effect*, which drives speakers and listeners to prefer the least confusable token of a category (useful in being clear), and an *articulatory effect*, which drives speakers to avoid articulatorily effortful tokens of a category (useful in being efficient). The least confusable token is likely to be a peripheral token in the category space, but such tokens also require more articulatory effort than more central realizations: thus, the two effects counteract.

All five levels in Figure 6 are relevant to B&H's explanation of auditory dispersion, which can be summarized as follows. The language-learning infant builds phonological categories around peaks in the distributions of her auditory input; on the basis of her lexicon, she establishes relations between these distributions and categories (lexicon-driven learning), i.e. she acquires an appropriate cue constraint ranking for her ambient language; subsequently, she reuses this ranking in production, now with an added articulatory effect. Put briefly, her strategy is: "optimise comprehension only, then just speak" (B&H: 256). In perception, some transmission noise is present (Ohala 1981), representing different kinds of variation and noise in communication: between-speaker variation, within-speaker variation, background noise and noise in the ear of the listener.

The prototype effect is embodied by the ranking values of the cue constraints: constraints militating against perceiving non-confusable tokens of a category have low ranking values, and since the BiPhon model is bidirectional, these constraints also favour the production of a category with a non-confusable auditory value. The articulatory effect is embodied by the ranking values of the articulatory constraints: these are highest for constraints militating against the production of peripheral auditory values. With computer simulations, B&H show that the interaction of the two effects yields optimally dispersed phoneme inventories. The output of a generation (represented by a single learner)[3] serves as the input to the next generation; already over a small number of generations, any input distribution will evolve into a stable inventory. Optimally dispersed initial distributions remain unchanged over the generations; in exaggerated initial distributions, where the category peaks lie more toward the edges of the auditory continuum, the articulatory effect forces the categories towards the centre of the phonetic space; in confusing initial distributions, where the category peaks lie close together, the prototype effect forces the categories to a sufficient mutual distance. B&H are also able to mimic the sound change in the history of Polish mentioned in §2 (p. 9).

In their formalization, B&H do not assume any faithfulness constraints to be violated, as they are concerned with single segments (although the ranking of faithfulness constraints does contribute to the prototype effect, for instance by preventing segments from the UF to undergo deletion or assimilation in phonological production (i.e. the mapping from UF to SF)), so that UF and SF are identical and the former can be disposed of. Also, as speakers are assumed to have perfect sensorimotor knowledge, AudF and ArtF are conflated into one Phonetic Form. This leaves the two-level production model of Figure 7:

/Surface Form/ ········ STRUCTURAL

CUE

[Phonetic Form] ········ ARTICULATORY

Figure 7. *Simplified production model (B&H: 227)*.

[3] The model was implemented into a more complex world of interacting agents by Van Leussen & Vondenhoff (2008).

B&H simulate the evolution of dispersion along a single auditory continuum, namely the spectral centre of gravity in sibilants; Boersma (2007, 2009a) did so for a two-dimensional space, namely the vowel space. In these simulations optimally dispersed inventories emerged as well, whose final distribution of categories was not influenced by the initial distribution, as it turned out to be in Liljencrants & Lindblom (1972): in fact, the first generation acquired a system with vowels that were random located in the auditory space. Van Leussen (2008) extended Boersma's (2007) formalization with populations of agents, thus combining Boersma's work with that of De Boer (1999, 2000, 2001), and found typologically realistic results for smaller vowel inventories.

4 Artificial neural networks

An artificial neural network or *neural net* is a network of interconnected single units or nodes, used to simulate different aspects of cognition, such as vision, category learning and stimulus classification. The application of neural nets is widespread: also outside the realm of science, they serve a wide variety of purposes, such as decision-making in stocks or sorting plastic in recycling plants.

4.1 Neurobiological underpinnings

The name 'artificial neural network' implies that the units in such a network correspond to neural structure in animals. To elucidate the analogy, I briefly introduce the structure and working of a typical mammalian neuron here. The notion of a 'typical' neuron is perhaps somewhat misleading, as the structures and workings of neurons show a great deal of diversity.

Neurons are the building blocks of the nervous system, and serve to process and transmit several types of information. Neurons can be classified into a number of categories: sensory or receptor neurons process and react to input from the sensory organs; motor neurons regulate muscle activity; and interneurons connect neurons to other neurons. Most neurons are located in the cerebral cortex. The number of cortical neurons varies greatly between animal species: mice have ca. $4 \cdot 10^6$ cortical neurons, cats have ca. $3 \cdot 10^8$, horses have $1.2 \cdot 10^9$ and humans have ca. $1.15 \cdot 10^{10}$ (Roth & Dicke 2005: 251). Estimates of the total number of neurons in humans lie in the order of magnitude of ca. 10^{11}

neurons (a.o. Azevedo et al. 2009; Williams & Herrup 1988). The number of synapses – the connections between neurons – in the human brain lies somewhere around 10^{14}, so a neuron has ca. 10^3 synapses on average (or even 10^4, according to Mitchell 1997: 82): the neural system is thus densely interconnected.

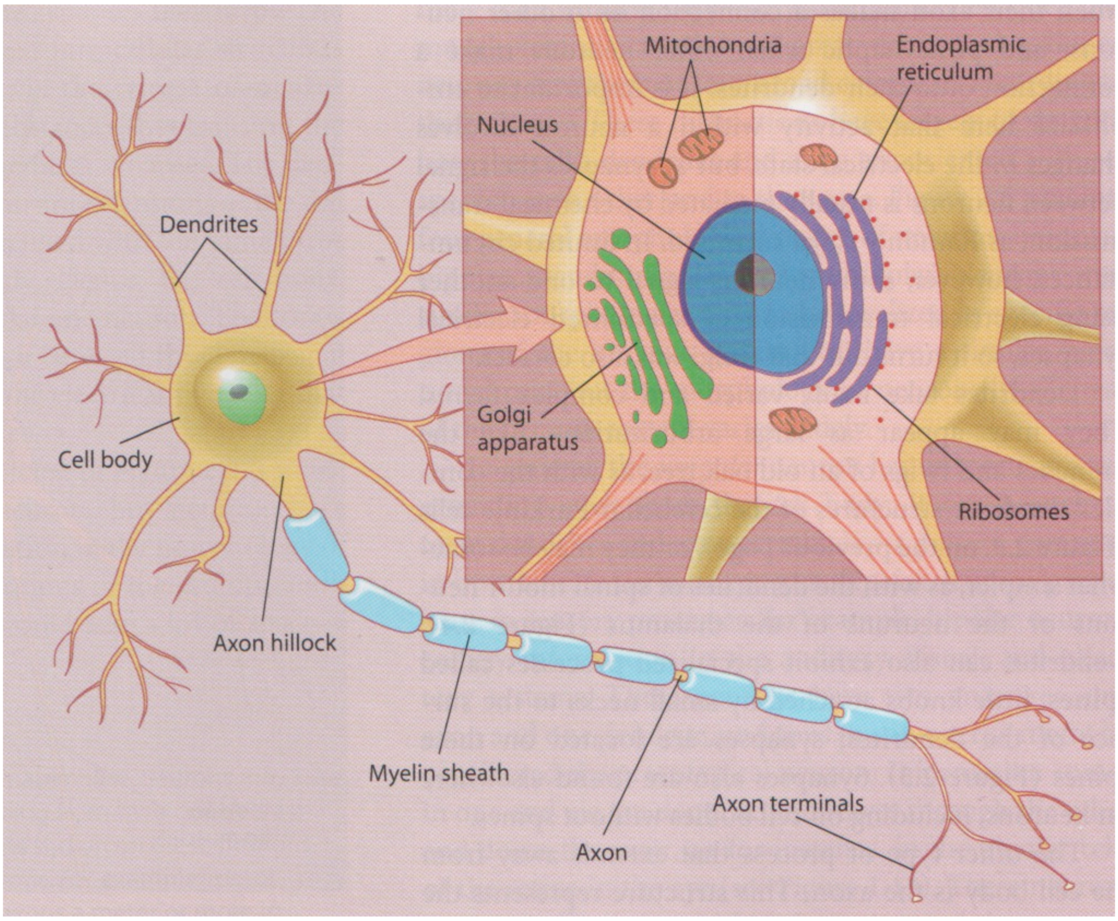


Figure 8. *A typical mammalian neuron*. *Image from Gazzaniga, Ivry & Mangun (2009): 21.*

A neuron consists of a cell body, or soma, from which two other parts emerge: tree-like structures called dendrites (or, somewhat redundantly, 'dendritic trees'), and an elongated structure called the axon. The axon ends in the axon terminal, where it branches out. The joint between this arborization and the dendrites of an adjacent neuron is the synapse or synaptic junction. It is here that information is transferred from one neuron to the next; these neurons are called presynaptic and postsynaptic, respectively. Within the neuron, information is transferred through the axon via electrical current (in many neurons, including the one depicted in Figure 8, the axon is covered in a myelin sheath that increases the speed of the impulse); this is impossible between neurons, as there is a small gap called the synaptic cleft. Here, at the synapse, information transmission occurs via chemical activity, viz. by neurotransmitter ions. These are enclosed in synaptic vesicles, which lie in the presynaptic axon terminal. The neurotransmitter ions are released into the synaptic cleft due to an action potential from the axon, and bind to receptors on the cell membrane of the postsynaptic neuron. The influx of ions generates a postsynaptic potential, which results in a change in membrane potential. The amplitude of this change defines the strength of the synaptic connection between the two neurons. The dendrites of the postsynaptic neuron conduct the electrochemical stimulation to the cell body, which integrates all incoming signals, and generates an action potential if a certain threshold is reached: then the neuron fires.

The synapse described here is excitatory, i.e. it increases the probability that the postsynaptic neuron generates an action potential. A synapse can be inhibitory as well, i.e. it decreases the probability of an action potential in the postsynaptic neuron by exerting a counteracting influence on its membrane potential.

4.2 Artificial neural nets

As written earlier, a neural net is an artificial network of interconnected units called nodes, which are grouped in layers. Nodes in adjacent layers, and sometimes also within layers, are linked by connections that may be either excitatory – i.e. if a node is activated, the activity in the connected node(s) increases as well – or inhibitory – i.e. if a node is activated, the activity in the connected node(s) decreases. The degree to which activity can spread from one node to another is determined by the weight of the connection between them. Most neural networks are used for purposes that require learning: this process generally involves changes in the weights of the connections (see §4.3).

In spite of their respectable age – the first neural net was created in the '40s of the previous century (McCulloch & Pitts 1943) – neural nets are not used very frequently in linguistics, where constraint grammars such as Optimality Theory and Harmonic Grammar (Smolensky & Legendre 2006) prevail. One of the principal advantages of neural nets over constraint grammars, however, is their biological validity. Both in animal cognition and in neural nets, different types of behaviour are basically different types of activation patterns: constraints only describe this behaviour, and hence are an abstract derivative at best. Another advantage of neural nets is the relative simplicity of the decision mechanism, which adds to their biological feasibility. In constraint-based grammars, by contrast, the evaluation process must take into account large lists of candidate input forms, and each additional level of representation in the grammar adds to the complexity. In neural networks this process is much more straightforward.

Also, neural nets are capable of modelling phenomena that reveal gradient properties of phonological categories; in constraint grammars, categories are discrete, so such grammars must fail. An example of such a phenomenon is the warping of the auditory space in the perceptual magnet effect (Kuhl 1992), modelled with neural nets by Guenther & Gjaja (1996) and within the BiPhon model by Boersma, Benders & Seinhorst (fc.). In an implementation of the BiPhon model with neural nets, the same representations are assumed as in Figure 6 (p. 14), and the description below this figure still applies: the word 'constraint(s)' should merely be replaced with 'connection(s)'. The bidirectionality of the model is warranted by making the connections symmetric, i.e. by making the weight of the connection from node A to node B identical to the weight of the connection from node B to node A (Hopfield 1982). The parallelism of the model is maintained because activation can flow through the network in both directions, simultaneously top-down and bottom-up.

4.3 Learning in neural nets

As mentioned earlier, learning in neural nets usually involves connection weight adjustment. This section (a brief summary of §4 of Boersma, Benders & Seinhorst fc.) discusses a number of weight update rules. One of the best-known rules is Hebbian learning; algorithms that are relevant to this thesis are the in- and outstar learning rules by Grossberg (1969) and the inoutstar learning rule by Boersma.

The general weight update rule (1) describes Δw_{ij}, the weight change of a connection from input node *i* with activity a_i to output node *j* with activity a_j:

(1) $\Delta w_{ij} = \eta_w(a_i a_j - \text{instar}\cdot a_j w_{ij} - \text{outstar}\cdot a_i w_{ij} - \text{leak}\cdot w_{ij})$

η_w is the learning rate, the amount by which a connection weight can change in every learning step: the neurobiological correlate of this parameter is synaptic plasticity.

One of the earliest proposed learning algorithms to be applied in neural nets is Hebbian learning (Hebb 1949), popularly summarized as “cells that fire together, wire together”: if two neurons happen to fire simultaneously, the strength of the connecting synapse increases, and their activities (or lack thereof) synchronize. If we are to cast Hebbian learning into equation (1), the terms instar, outstar and leak are all zero, so $\Delta w_{ij} = \eta_w a_i a_j$. An undesirable consequence of this strategy is that the connection weights eventually become unlimited. This can be resolved in a number of ways.

One strategy is to introduce leak in the connection weights (in (1), the term leak ≠ 0, the other terms are zero); if leak = 1, a connection weight will come to be the probability that its input and output nodes are on simultaneously. This means, however, that connections to nodes that are rarely active will always be weak, even if the correlation between the activities of the nodes is strong, so that processing is determined by frequency rather than reliability.

In outstar learning (outstar = 1, the other terms are 0), the weights come to be the probability that the output node is on, given that the input node is on (which is analogous to the way OT handles inputs); this resolves the low-frequency problem, because it focuses on the predictability of the output from the input. Instar learning (instar = 1, the other terms are 0), as the name suggests, works in the opposite direction: with this algorithm, the weights come to be the probability that the input node is on, given that the output node is on.

Both out- and instar learning are directional, which is troublesome if we want to use connections bidirectionally. Inoutstar learning (instar = outstar = 0.5, leak = 0) combines the two: the expectation values of the connection weights in an inoutstar network are the harmonic means of these weights in an outstar- and in an instar network. The inoutstar learning rule is insensitive to the direction of processing, and thus best suited for bidirectional use: it optimizes both production and perception, but not perfectly. Nevertheless, it solves some of the problems associated with the in- and outstar algorithms.

PART II: APPLICATION

5 Auditory dispersion in neural networks

This paragraph provides descriptions of the neural network used in the computer simulations, and of the learning procedure. All simulations were run in PRAAT (Boersma & Weenink 2012, version 5.3.19). Separate scripts were used for simulations of the evolution of unimodal and bimodal distributions (Appendices A (p. 70) and B (p. 76) respectively; updated to be used with PRAAT 5.3.23, the latest version when this thesis was finished).

5.1 The network

The script creates an artificial neural network consisting of three layers of nodes, each corresponding to a level of representation in Figure 6 (p. 12): /SF/, [[AudF]] and [ArtF]. Each node on the SF layer represents a discrete phonological category; each node on the AudF layer represents a frequency value (or, more precisely, a range of values) on an auditory continuum. Note that the AudF layer could represent any auditory continuum (F_1, F_2, VOT, centre of gravity, etc.), and that an SF node could represent any category that corresponds to a range of values on an auditory continuum, such as a feature or segment. For instance, [+velar] could relate to low F_2 values, and [–velar] to high F_2 values; or /b/ to negative VOT values, /p/ to VOT values around zero and /p^h/ to positive VOT values. In the remainder of this thesis, however, I will mostly be speaking in general terms, i.e. about categories.

The script 'asks' the user to enter the desired number of AudF nodes: the default value of this number is 100. This number was chosen because it was thought to strike a reasonable balance between accuracy (i.e. it made the range of frequency values that each AudF node represents fairly small, and allowed for a precise computation of the basic statistics) and computational feasibility.

A network contains only one ArtF node, with inhibitory connections to every AudF node, realizing the articulatory effect. These connections do not learn. Their weights are determined by the location of the connected node on the AudF layer: nodes at the extremities of this layer represent peripheral values of the auditory continuum, which require most articulatory effort and hence are inhibited most strongly by the ArtF node. The connections to the central nodes on the AudF layer, on the other hand, are weaker, as these nodes correspond to less effortful articulations. Note that these connections are not sensorimotor connections: as in B&H, sensorimotor knowledge is assumed to be perfect. Rather, this strategy allows for a more transparent visualization of the interaction of perceptual and articulatory drives. Implementation of sensorimotor connections is of course perfectly possible: it would entail as many ArtF nodes as there are AudF nodes, each connected to its corresponding AudF node and the adjacent ArtF nodes to allow mutual inhibition.

The weight w of the connection between the ArtF node and the xth AudF node is determined by a parabola described by the function $w = c - d(x - m)^2$, where c is the connection strength from the ArtF node to the central AudF node (i.e. the vertical displacement of the parabola), d is a measure of the degree of inhibition, and m is the number of the central AudF node. The central AudF node is the node that lies exactly halfway the AudF continuum; its number is only an integer if the layer contains an odd number of nodes. For instance, if the layer has 100 nodes, $m = 50.5$; if the layer has 101 nodes, $m = 51$. The script asks the user to enter the vertical displacement of the parabola, say c, and a parameter indicating the degree of inhibition, say i: the value of w at the edges of the AudF layer equals $c - i$. To achieve this, d must be $4i/(n^2 - 1)$, with n being the number of nodes on the AudF layer. Figure 9 shows a number of such parabolas for different values of i. In all cases, $c = -0.2$.

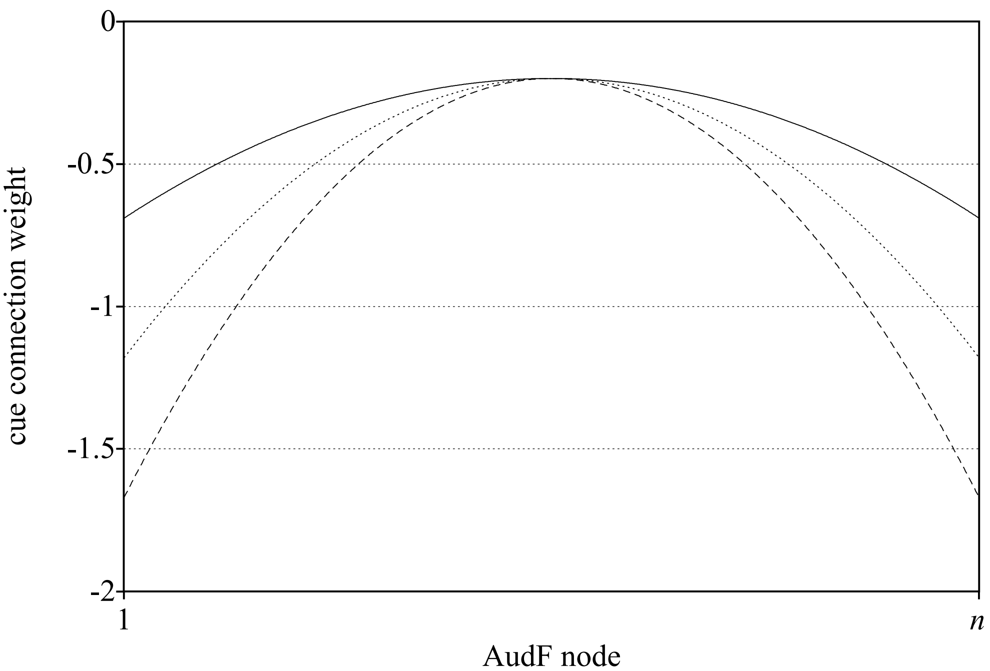


Figure 9. *Articulatory inhibition parabolas for different values of* i *(solid line:* i *= 0.5; dotted line:* i *= 1; dashed line:* i *= 1.5). In all cases,* c *= -0.2.*

All SF nodes are connected to all AudF nodes with cue connections. In the initial state of the network, the SF and AudF nodes have a random activity (value between 0 and 1), and the cue connections have a small random weight (value between 0 and 0.25). The ArtF node has an activity of 1; as mentioned above, the weights of the articulatory connections are fixed and depend on the number of categories in the system. This is a simplification, because the inhibitory connections will become weaker in those areas where the learner gains sensorimotor knowledge by experience. This means that the articulatory weights are different in systems with more categories, which will be implemented in some of the simulations in §6.

In its initial state, the network looks as in Figure 10 (here with 2 category nodes and 50 AudF nodes). Red nodes have a positive activity, whose size is indicated by the diameter of the circle; nodes with a negative activity are drawn in blue (although not (yet) in Figure 10). Excitatory connections are drawn in black, inhibitory connections in white. The weight of a connection is indicated by the width of the line, as can clearly be seen in the articulatory connections: along the auditory dimension, the widths of the lines drawing the articulatory connections reflect the parabolic shape of the effort curve.

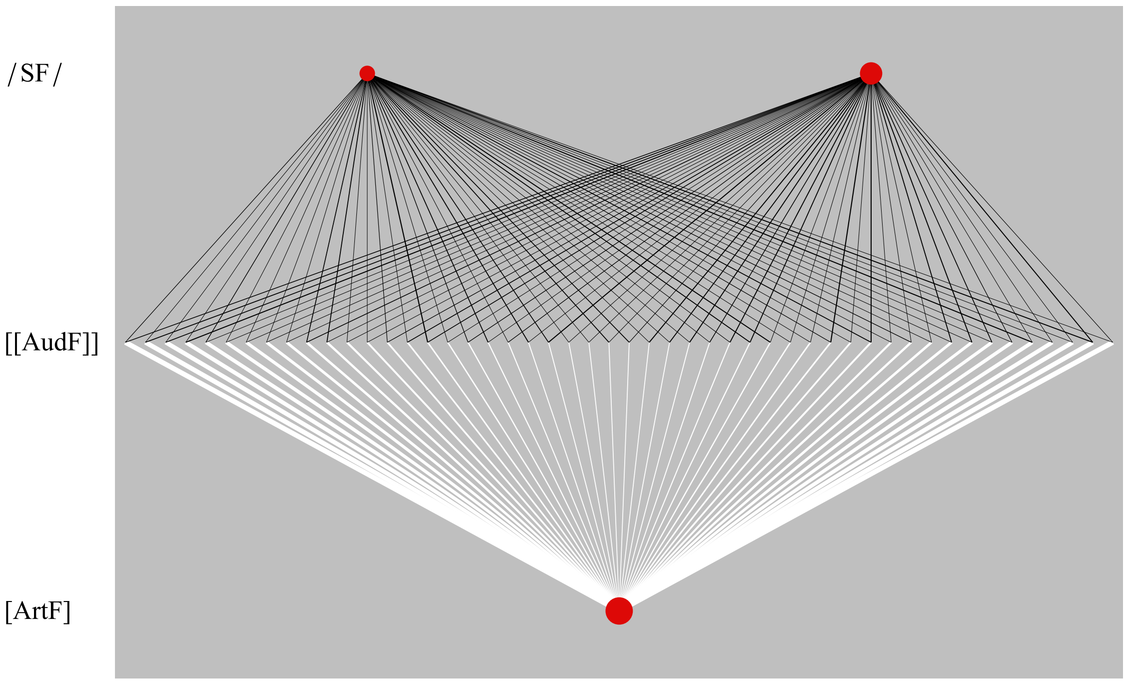


Figure 10. *The initial state of a network with two categories.*

5.2 Learning and transmission

The script produces a chain of virtual learners who learn from the output of the previous learner, reminiscent of the process of iterated learning (Kirby 2001). Both B&H's strategy and iterated learning expose human biases through repeated interactions. Iterated learning investigates higher-level inductive biases: learners are only exposed to a subset of the output of the previous generation, and have to form a hypothesis about the structure underlying this output. B&H are concerned with lower-level perceptual and articulatory biases, so learners can learn from the full output of the previous generation. In the simulations in §6, as in B&H, each generation is represented by a single speaker (unlike in Van Leussen 2008 and Van Leussen & Vondenhoff 2008).

The locations of the category peaks in the initial distribution are entered by the user. For each combination of phonological category and auditory value, a probability is computed: these probabilities are normally distributed along the auditory dimension. The probabilities are stored in a table. Each category has the same standard deviation, which is equal to 1/16 of the continuum (in B&H: 1/14.2).

Before every learning step, all activities in the network are set to zero. The ArtF node is inactive. Then one of the categories is selected at random; for this category, a frequency value on the auditory continuum is chosen from the table (i.e. a node on the AudF layer). The probability with which a value or node is chosen reflects its probability in the output distribution of the previous generation: more frequent outputs appear in the input of the next generation more often. The intergenerational transmission is a little noisy: the frequency perceived by the listener is drawn from a normal distribution whose mean is the produced frequency, and whose standard deviation is 5% of the number of AudF nodes (as in B&H). This number is not rounded off to the nearest AudF node.

Subsequently, the SF node of the selected category is switched on, i.e. its activity is set to 1. A normal distribution of activities is set up around the perceived frequency node (see also §6.6). This Gaussian curve reflects the excitation spreading on the basilar membrane in the human ear, where an incoming sound wave does not only excite the hair cells sensitive to that frequency, but also – to a lesser extent – adjacent hair cells. The standard deviation of this normal distribution does not depend on the number of phonological categories. This spreading is illustrated in Figure 11, showing the activities along an AudF layer with 100 nodes with an incoming frequency of node 50; the standard deviation is 5% of the number of AudF nodes. The activities of the nodes on the AudF layer are set according to this curve.

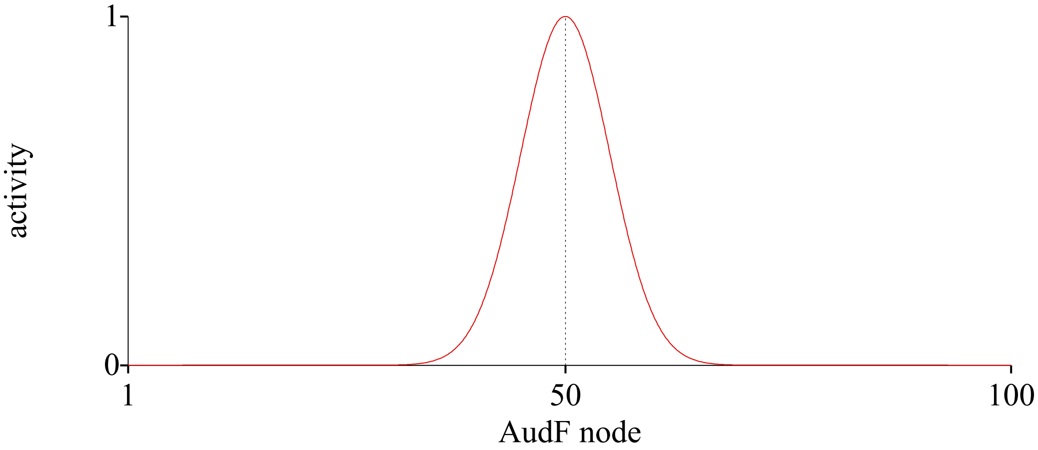


Figure 11. *Activity spreading on the AudF layer around the perceived frequency.*

Then all SF and AudF nodes are clamped, i.e. held fixed, and the weights of the cue connections are updated by a small amount (the learning rate, §4.3). The learning of the appropriate weights corresponds to the learning of the appropriate cue constraint ranking in B&H. To model perceptual learning in the perception direction, we can use inoutstar learning, which can be used in both directions of processing, or outstar learning, which – just as OT – is output-oriented and thus expected to replicate the Optimality-Theoretic foundation of B&H.

All connections are bidirectional, so the speaker uses the same connections in production. Additionally, in this direction, the ArtF node is switched on, so that the inhibitory articulatory connections come to constrain the activities at the AudF layer. In principle, the activity of an unclamped node i is the sum of the weights of every connection w_{ij} leading to it times the activity of node j from which this connection emerges: $a_i = \Sigma a_j \cdot w_{ij}$. Additionally, it is possible to clip the activities of unclamped nodes. PRAAT offers three possibilities: linear clipping between two values $a_{\min}$ and $a_{\max}$, e.g. 0 and 1, which means that any activity below $a_{\min}$ or above $a_{\max}$ is set to that value; top-sigmoid clipping, in which larger activities smoothly approach $a_{\max}$; and sigmoid-clipping, in which smaller activities smoothly approach $a_{\min}$, and larger activities smoothly approach $a_{\max}$.

Figure 12 shows the activities in the network from Figure 10 after 50,000 learning steps of the outstar algorithm (learning rate 0.001), producing the left category. (Hereafter I will refer to the categories in a two-category network as /A/ (left-hand category) and /B/ (right-hand category).) The figure also indicates the prototype (the AudF node with the strongest connection to that category node; left arrow) and the node with the highest output activity (right arrow) – this is why I took 50 nodes instead of 100: this way the separate nodes are still visible. Because of the articulatory effect, the most active node is inclined more towards the centre of the AudF, where the inhibitory connections are weaker.

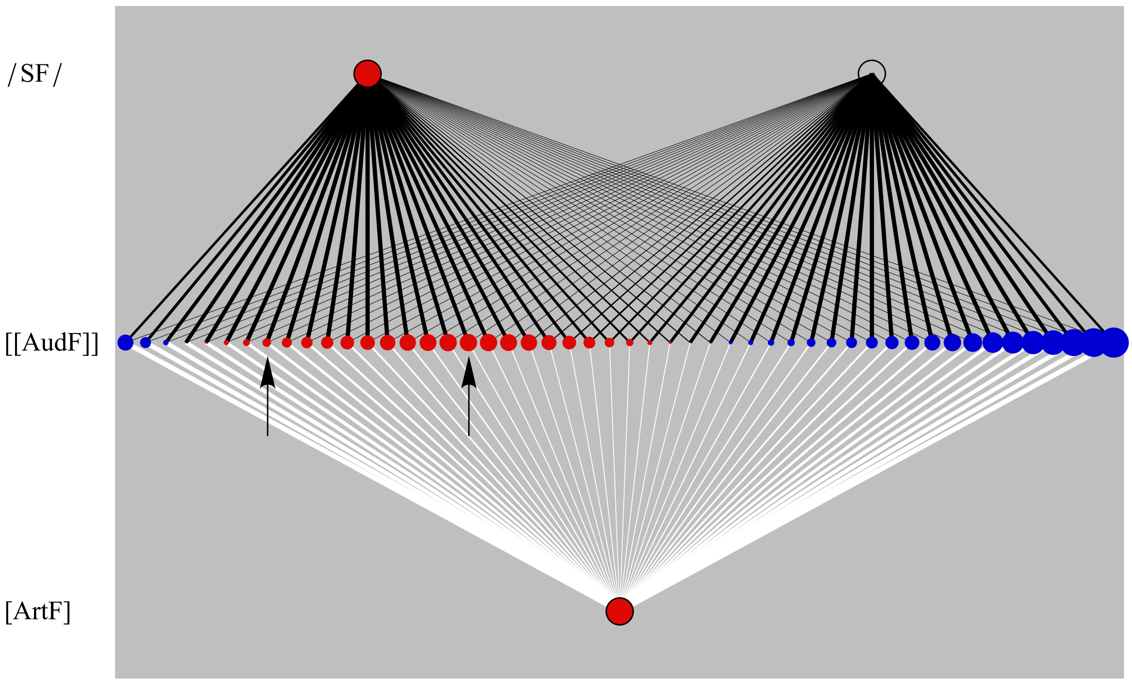


Figure 12. *The network from Figure 10 after 50,000 learning steps (outstar update rule), producing category /A/.*

A network with inoutstar learning will come to look quite different from Figure 12. Inoutstar learning is less sensitive to the predictability of the output from the input than outstar learning; the effect of this is seen most clearly at the edges of the auditory dimension, where the frequency values have the lowest probabilities. Because the cue connections at the edges are weaker, the AudF nodes there experience more inhibition from the ArtF node. This can be seen clearly by the fact that there are more nodes drawn in blue, and that these discs have larger diameters:

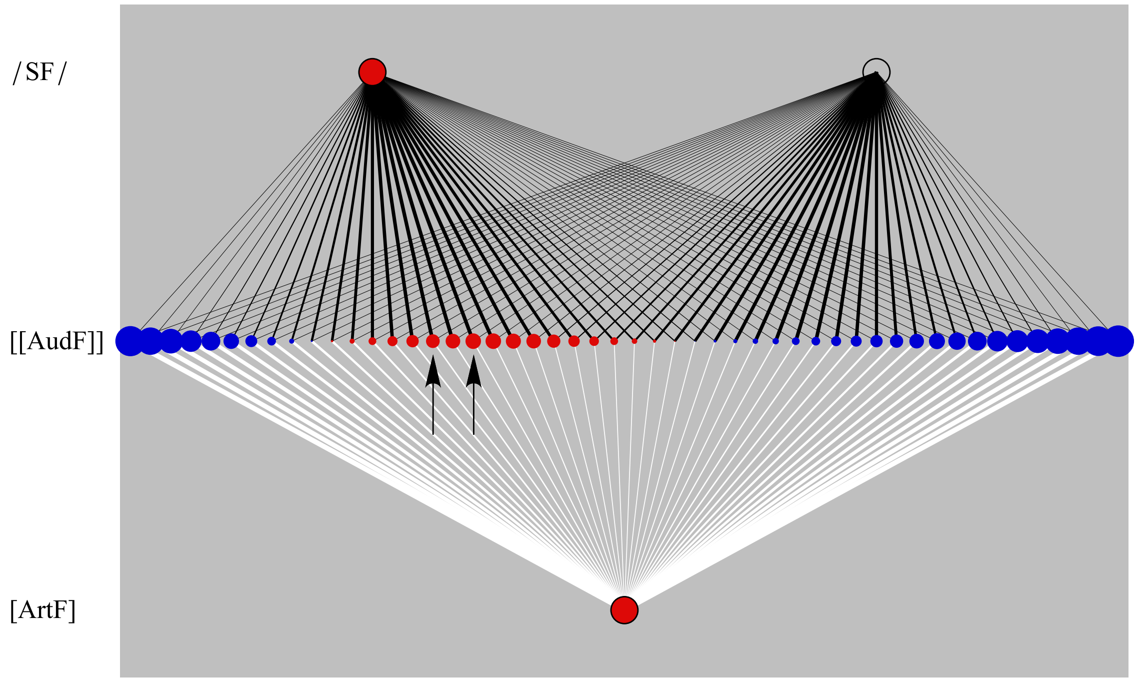


Figure 13. *An inoutstar network after 50,000 learning steps. The left arrow indicates the prototype, the right arrow the most active node.*

In order to convert the activity a of an AudF node into its output probability p, we can use a number of formulas. The activity of a node can be interpreted directly as its probability (a linear activity-to-probability rule), in which case the activities cannot have negative values; another possibility is to use the formula $p = e^{a/T}$ (an exponential activity-to-probability rule) where T is the temperature, a parameter indicating the evaluation noise (Ackley, Hinton & Sejnowski 1985). In both cases, the relative output probabilities are computed by dividing every p by the sum of the p values of all AudF nodes for a category. These probabilities are stored in a table, which serves as the input to the following generation. Then the whole process starts anew.

Note that in Figure 12, there still exist (weak) connections from both category nodes to the opposite extremes of the AudF layers (their weakness is less clearly visible in the pdf version of this document): these values occur in the input only very infrequently and hence contribute very little to the learning process. The weights of these connections thus differ very little from their initial values.

6 Simulations

This chapter presents the results of the computer simulations. Comparisons will be drawn between simulations with outstar learning and simulations with inoutstar learning (§6.1–6.4). The outstar algorithm, because of its similarity to OT, is likely to replicate B&H's results, while the bidirectionality of the inoutstar algorithm is expected to make it a good candidate for various types of phenomenon. This algorithm has proven capable of mimicking another (unrelated) phenomenon in the phonology-phonetics interface, the perceptual magnet effect (Boersma, Benders & Seinhorst fc.).

All simulations in the following sections make use of an exponential activity-to-probability rule with top-sigmoid activation clipping: this provided the smoothest output probability curves. Other possibilities, already touched upon in the previous chapter, are discussed in §6.6.

I investigate the evolution of auditory dispersion in different types of categories: two-category inventories with contiguous categories (§6.1a, 6.2a), three-category inventories (§6.1b, 6.2b; these sections include simulations of the sound change in the Polish sibilant system described in §2, p. 7), one-category inventories (§6.1c, 6.2c) and two-category inventories with non-contiguous categories (§6.1d, 6.2d). The evolution of systems without articulatory limitations is investigated in §6.3; a comparison between outstar and inoutstar learning is drawn in §6.4; the auditory dispersion of the Dutch sibilant system is treated in §6.5; the parameter settings of the network are discussed in §6.6.

6.1 Outstar learning

6.1a Two-category inventories (with contiguous categories)

In this section I treat three types of initial distributions: standard; exaggerated; and confusing and skewed. The standard distribution is defined as a distribution with peaks in the input at 35% and 65% of the continuum, where they lied approximately in B&H (§5.7). The initial probability curves (or production probabilities of "generation 0") for this type of distribution are drawn in Figure 14. These probabilities have been calculated by the script, i.e. the distributions are bell-shaped because they were computed with a formula for a Gaussian distribution.

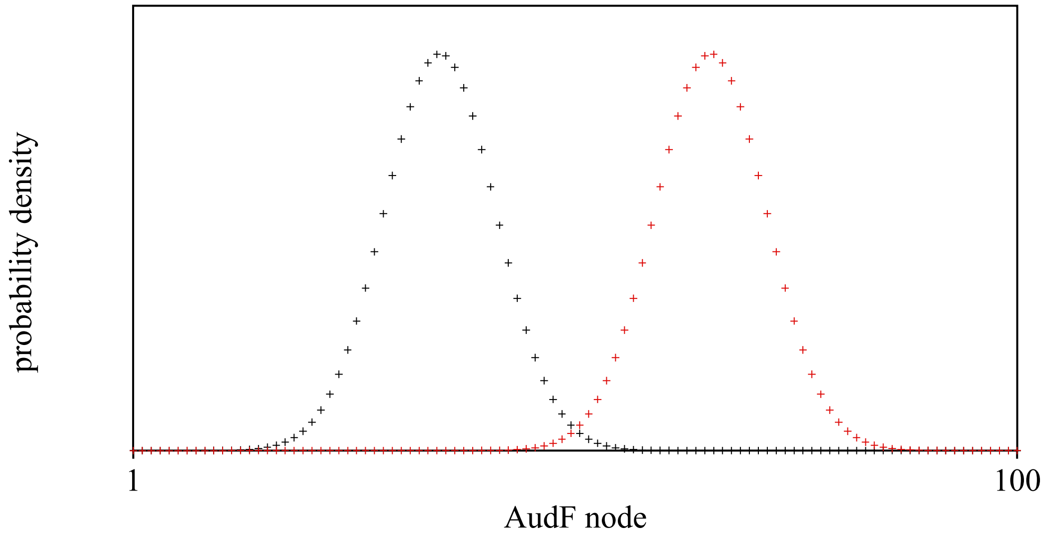


Figure 14. *Standard initial distribution (output probabilities of "generation 0").*

The production of generation 1 is shown in Figure 15. These probabilities are the output of the grammar itself, i.e. their distributions are bell-shaped (although not symmetric) due to the interaction of the parameters in the network. These are: articulatory inhibition = 1.5; weight of central articulatory connection = -0.25; temperature = 0.1; transmission noise = 0.05; 100,000 tokens; learning rate = 0.001.

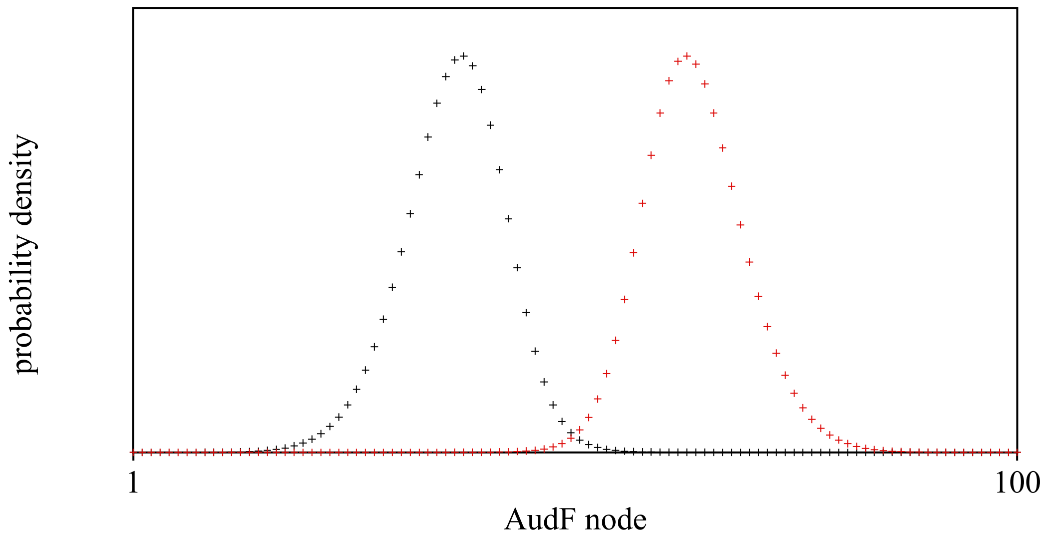


Figure 15. *Output probabilities of generation 1.*

Figure 16 shows the evolution of this inventory over ten generations. The black curve connects the average produced nodes, the grey curves connect the values of the average ± 1 standard deviation.

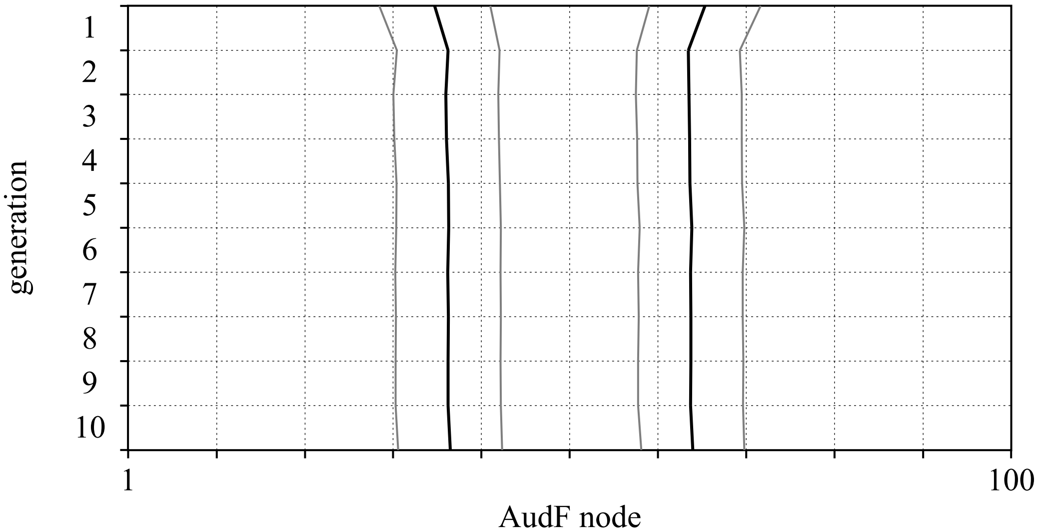


Figure 16. *The evolution of a two-category inventory (standard initial distribution, outstar learning).*

This distribution turns out to be stable over the generations: within one generation, the average produced node of a category has reached its target position. For category /A/, this lies at ca. 36% (node 37) of the continuum, for /B/ at ca. 64% (node 64) (B&H: 33.8 resp. 66.3%). As mentioned in §5.2, the initial standard deviation in a network with *n* AudF nodes is *n*/16 nodes, so in the present network it is 6.25 nodes. This is the standard deviation of the distribution that "generation 0" *produces* (Figure 14); due to the transmission noise, this distribution is slightly scattered, so that generation 1 *perceives* a larger standard deviation of 8.0 nodes. In its production, this generation reduces the standard deviation to 5.8 nodes, due to the articulatory effect (Figure 15): this is the entrenchment also found by B&H (p. 245). Generation 2, however, perceives this distribution with an increased standard deviation of 7.7 nodes because of the transmission noise. In subsequent generations, the standard deviations of the input and output distributions remain stable, because the entrenchment of the articulatory effect and the scattering of the transmission noise are in balance.

A highly exaggerated contrast, with peaks at 20 and 80% of the auditory continuum, is diachronically unstable and evolves into the equilibrium seen above within three generations. The first generation contributes most to this change by immediately moving the categories to more central values: apparently their articulatory effect outweighs their prototype effect by far. This can be seen in Figure 17:

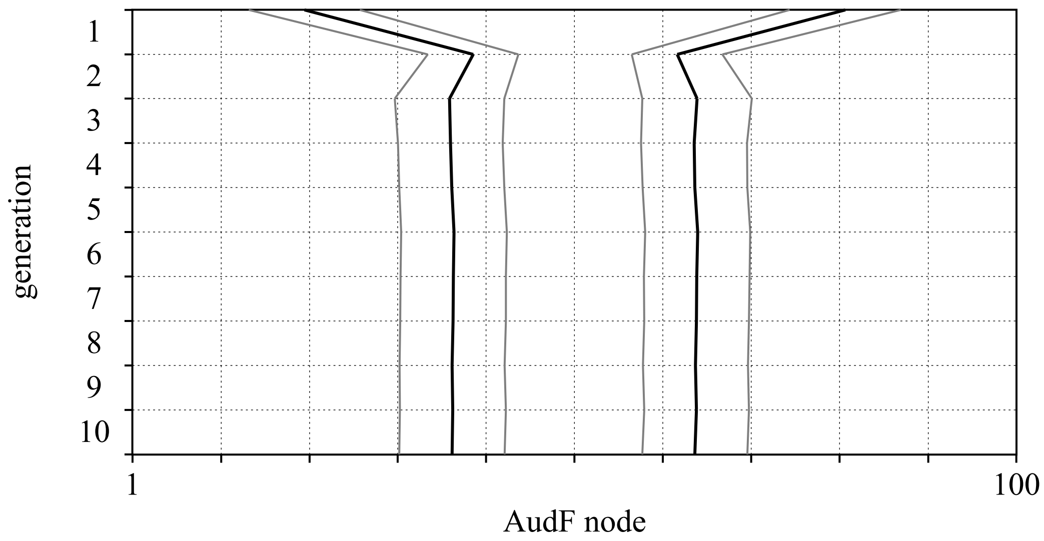


Figure 17. *The evolution of a two-category inventory (exaggerated initial distribution, outstar learning)*.

Surprisingly, and in contrast with B&H's simulations (p. 248), the first generation produces more central values than following generations – their articulatory effect is a little *too* strong, it seems. Closer scrutiny of the network of the first generation reveals that some cue connections remain from SF node /A/ to the right edge of the AudF layer, and from node /B/ to the left edge of the AudF layer. This is unexpected, because these are exactly the auditory regions in which no perceptual confusion should exist, since the peaks in the input distributions lie here. These inexplicable cue connections account for the shift of the distributions going inward to a large degree, as they move to the region of minimal confusability. Indeed, decreasing the articulatory inhibition does not solve the issue, so the cause lies in the perceptual learning process. The problem can be solved by increasing the learning rate by a factor 10 and decreasing the number of learning steps by a factor 10: however, this yields more fluctuation throughout the evolution of the system (cf. §6.6).

Figure 18 shows the evolution of a confusing and skewed distribution, with categories that lie close together and are not equidistant from the centre of the auditory continuum (their peaks lie at 50 and 65%, respectively).

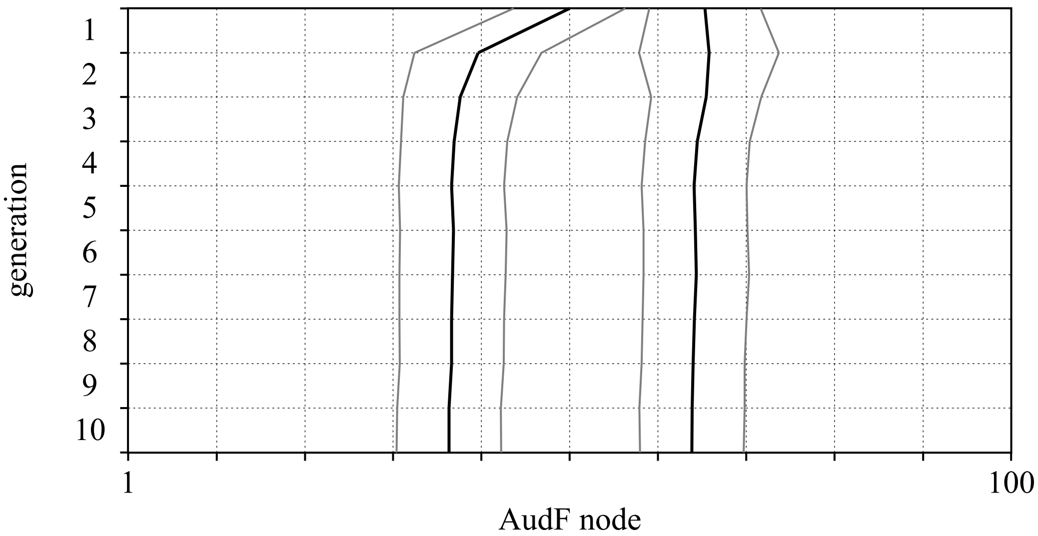


Figure 18. *The evolution of a two-category inventory (confusing and skewed initial distribution, outstar learning).*

The first generation directly enhances the contrast in the system: category /A/ is shifted into a lower auditory region, so that the between-category distance is the usual 27 nodes. The prototype effect thus acts very quickly. When the distance between the categories is sufficiently large, /B/ also lowers slightly, and both categories drift to the locations where they stabilized in the previous systems. The within-category variation increases in generation 1 because of the activity spreading along the AudF layer, which enlarges the auditory region associated with a category (cf. §6.6). The standard deviations already decrease in generation 2, because generation 1 has resolved the between-category overlap. Within ca. five generations we have the stable system familiar from Figures 16 and 17 (p. 30–31).

The evolution of a confusing and skewed distribution in B&H's simulations is slower, even though their distribution is less confusing than the one chosen here (peaks at 50 and 75% of the auditory continuum): optimal dispersion is only reached in some 15 generations, instead of 5 here.

6.1b Three-category inventories: Polish sibilants

This section treats inventories with three categories by providing a simulation of the evolution of the Polish sibilant inventory (cf. §2, p. 7).

In §5.1 (p. 23) I briefly mentioned that different numbers of categories in a system may require different articulatory settings: the more categories a language has, the larger the phonetic space these categories occupy, and the weaker the peripheral articulatory connections may have to be. Figure 19 shows what happened in the first ten generations of my simulations of the evolution of medieval Polish, where the same articulatory settings were used as in the two-category outstar simulations (peaks at identical locations as in B&H: from left to right: at 34.4% for /A/ [ʃ], at 64.4% for /B/ [s^j] and at 66.88% for /C/ [s]):

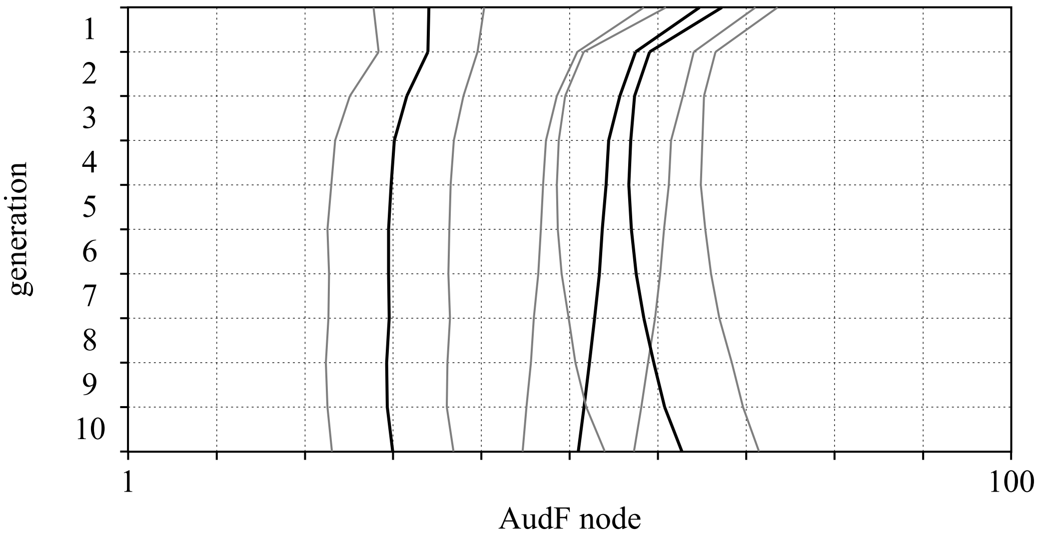


Figure 19. *The evolution of medieval Polish, using the articulatory settings from the two-category simulations (outstar learning).*

It seems that the articulatory effect is too strong again, because the prototype effect is nullified in the first generation. We would expect the confusing opposition [s^j] vs. [s] to be phonetically enhanced immediately, but it is not resolved at all: the average produced nodes of these categories are in fact closer together than they were in the initial distribution, and have just shifted to more central values due to the articulatory effect. This indicates that the effort curve may have to be less steep in the three-category simulations than it was in the two-category cases. In Figure 20, the degree of articulatory inhibition is set to 1.0 rather than 1.5; the weight of the central articulatory connection is kept at -0.25, because I assume that central articulations are equally lazy for all speakers, regardless of the number of categories in their system.

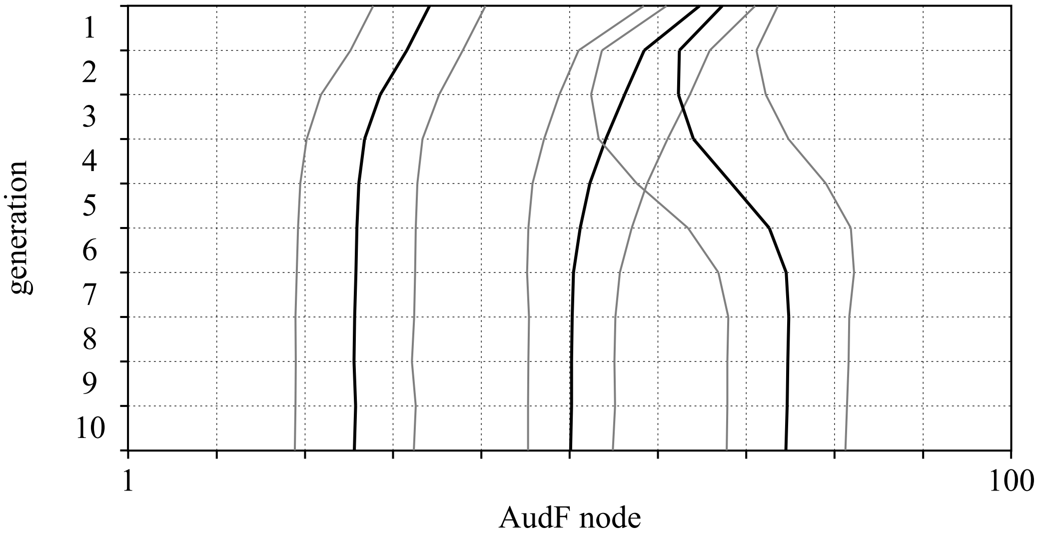


Figure 20. *The evolution of medieval Polish, with a less steep articulatory curve (outstar learning).*

These settings yield better results, although the [sʲ]–[s] contrast is still less enhanced in the first generation than might be expected: the articulatory effect still forces both categories into a less effortful region. Category /A/ ([ʃ] → [ʂ]) is pushed into a lower auditory region immediately, /B/ ([sʲ] → [ɕ]) lowers towards the centre of the continuum and /C/ ([s] → [s̪]) soon moves away from /B/ into a higher region. In the final state, the average produced node of /A/ lies at 25.4% of the continuum, that of /B/ at 50.0% and that of /C/ at 74.6% (B&H: 27.8, 50.5 and 72.4%). The entire system thus occupies a larger part of the auditory dimension than a two-category system (cf. Figure 4, p. 6): the difference between the average produced nodes of /A/ and /C/ in Figure 20 is 48.8% of the continuum, that between /A/ and /B/ in Figure 16 (p. 30) is 27.5%.

The reader might object that the articulatory curve should be shallower still in order to replicate the enhancement of the confusing initial opposition that B&H found (Figure 16, p. 250). However, it turns out that this strategy would not solve the problem: in fact, it would yield inaccurate auditory values in the final state, and predict output probability curves with relatively large probabilities for peripheral auditory values (i.e. asymmetric distributions with larger standard deviations). A simulation in §6.3 will show that even if the articulatory connections are disabled altogether, no major enhancement occurs in the first generation; a simulation in §6.6 will show that a smaller standard deviation of the activity spreading along the AudF layer reduces the problem (remember also the confusing distribution from §6.1a).

6.1c One-category inventories

According to B&H (p. 252), the first-generation learner of a one-category inventory will always produce a central auditory value, irrespective of the initial distribution. In their formalization, all cue constraints are ranked at the same height initially, and this ranking does not change during learning: after all, there is only one category, so every incoming sound will be perceived as this category. Because the learner does not make any mistakes, she does not have to rerank any cue constraints, and her output depends solely on the ranking of the articulatory constraints, which favours the auditory value at the valley of the articulatory curve.

In the present neural net, by contrast, the cue connections initially have random (i.e. usually unequal) weights, because not all synapses in the newborn's brain start out with identical strengths. This means that perceptual learning must take place in order to achieve a realistic output distribution. In an outstar network, only those cue connections learn that receive input; if the initial distribution has a peak at, for instance, 10% of the continuum, the cue connections at the right-hand end of the network will only become stronger after an extremely large number of learning steps, when the tokens at the right-hand periphery of the AudF layer (that have a low probability) have appeared in the input sufficiently often. Hence, the learner needs a large number of learning steps to achieve a symmetric network, or else the peak in the output distribution is most likely to lie at a non-central auditory value.[4]

Figure 21 shows the evolution of a one-category inventory with a skewed initial distribution. A steeper articulatory curve was chosen than in the two- and three-category simulations: the degree of inhibition is 2 (in the two-category simulations: 1.5, in the three-category simulations: 1). The figure shows that after 100,000 learning steps, generation 1 on average does not yet produce the central auditory value; this is only achieved by generation 2.

[4] With inoutstar learning and an extremely skewed input distribution, the peak in the output distribution of a single-category inventory will never move to a central auditory value within a single generation, because the cue connection weights will be more faithful to the input distribution than they are with outstar learning. See Figures 31 (p. 43), 43 (p. 53) and 44 (ibid.).

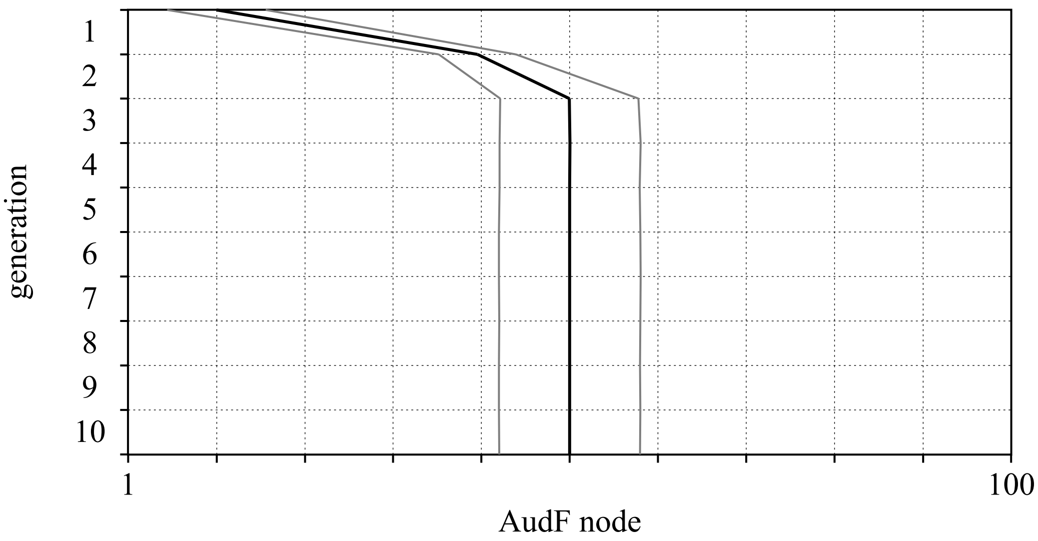


Figure 21. *The evolution of a skewed one-category inventory (outstar learning).*

The prediction that the first generation does not yet produce an central average frequency value even after 100,000 learning steps differs greatly from B&H's prediction: in their simulations, a learner of a single-category inventory already has an optimally dispersed system after having been exposed to only one token (or in fact zero, because the lexicon is already in place). They acknowledge that this "predicted one-shot shift to the centre" (p. 252) may strike some readers as unfeasible, but remark that it is difficult to test. The same is true of my prediction, so it cannot be established which is more realistic.

6.1d Two-category inventories with one non-contiguous category

The BiPhon model could, in principle, represent any language with non-contiguous categories, i.e. categories whose distributions have more than one peak. B&H argue that such languages, although representable, are diachronically unstable: they show that an inventory with one bimodal category (i.e. a category whose probability distribution has two peaks) and one unimodal category evolves into a system with two unimodal categories (§6.4, p. 252–254).

However, in this simulation one of the peaks of the bimodal category is smaller than the other: the probability of the peak located at 25.0 Erb is three times smaller than the probability of the peak at 31.0 Erb. Such a distribution is shown in Figure 22:

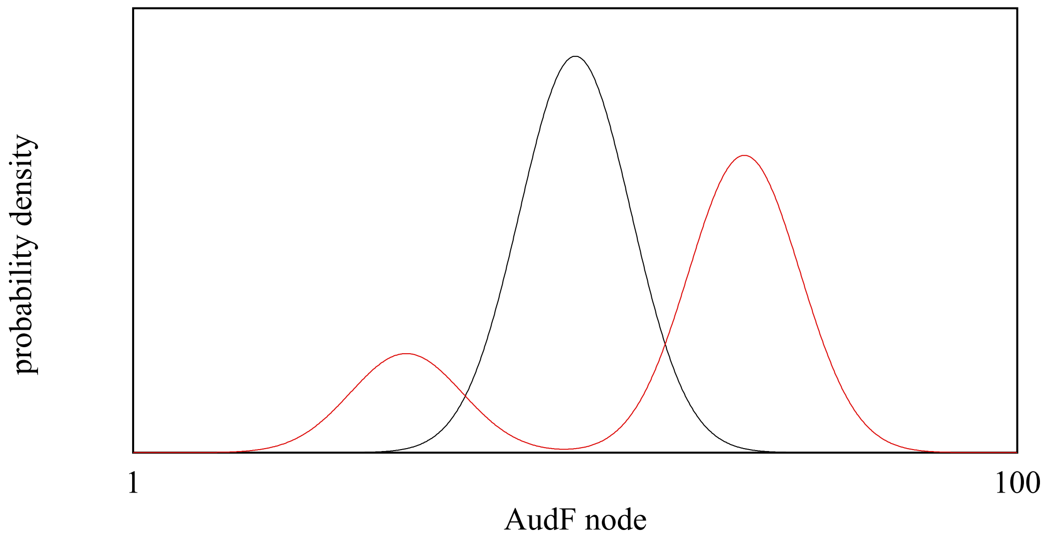


Figure 22. *An inventory with one bimodally distributed category. The probability of the right-hand peak is three times higher than that of the left-hand peak.*

Because the distribution from Figure 22 is biased towards one side of the continuum, the smaller peak may be more prone to demise than the larger peak from the very start, since speakers have learned in perception to prefer the less confusable tokens of a category. This means that the emergence of optimal dispersion in B&H's simulation of a bimodal distribution is perhaps due to the chosen probability curve, rather than being a virtue of the model itself. To answer this question, I also simulate the evolution of a distribution in which both peaks of the discontiguous category have the same probability, as in Figure 23:

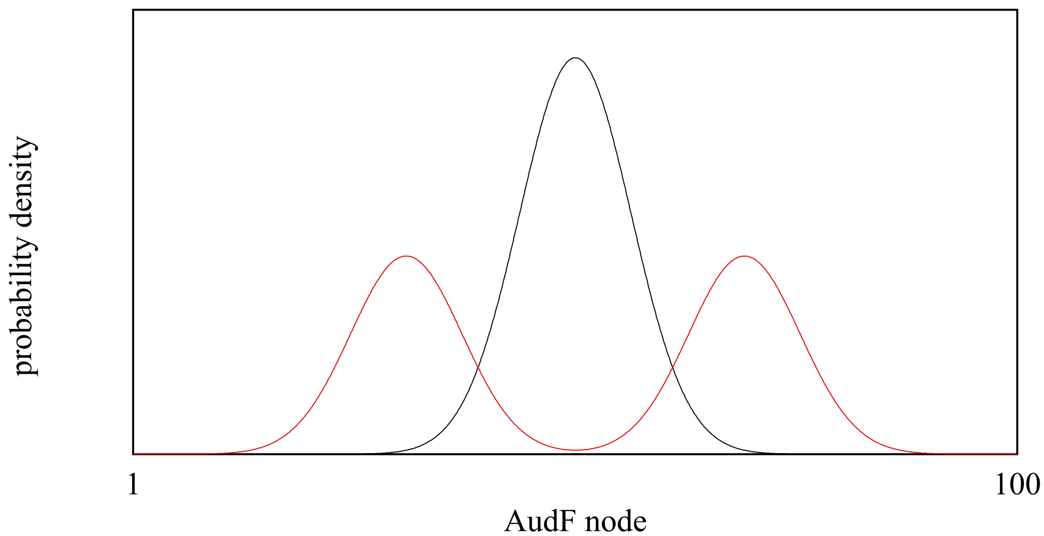


Figure 23. *An inventory with one bimodally distributed category. Both peaks have equal probabilities.*

The evolution of the distribution from Figure 22, with an asymmetric bimodal category, in an outstar neural net is shown in Figure 24 (as in the previous two-category simulations, the articulatory inhibition was set to 1.5):

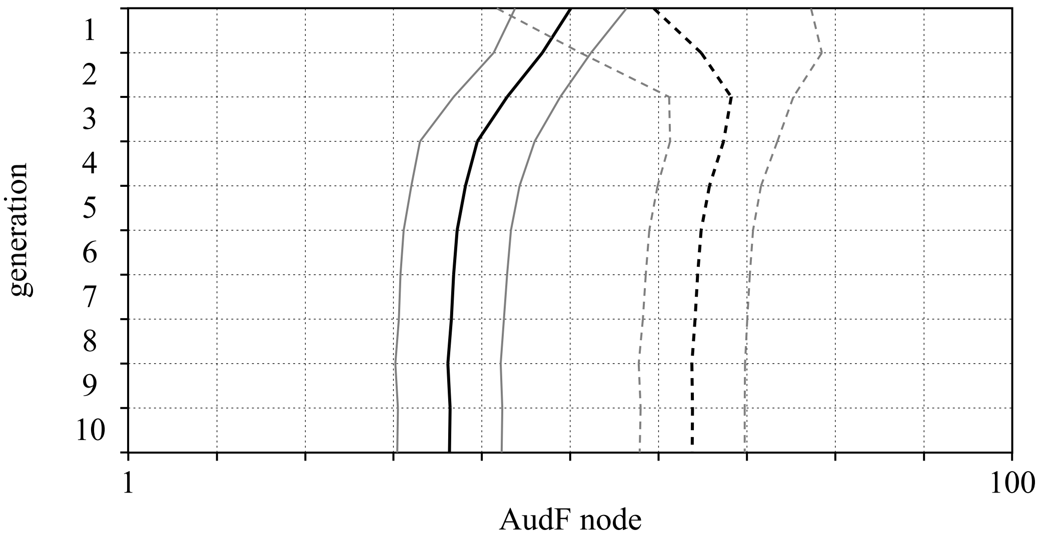


Figure 24. *The evolution of the inventory from Figure 22: bimodal category drawn with a dashed line (outstar learning).*

This inventory turns into the well-known optimally dispersed inventory, and has already reached the stable state within ten generations instead of the nearly twenty that were needed in B&H. The gradual disappearance of the smaller peak of the bimodal category is signalled by the normalization of the standard deviation of this category; this is visualized in a more tangible manner in Figure 25, showing the output probability curves of “generation 0” and the first four generations of learners. Because of the perceptual bias towards the right peak, generation 1 already strongly prefers this peak; it has moved slightly towards the edge, because the unimodal category lies in the middle of the continuum and tries to push the peaks of the bimodal category away. The left peak of this category has already begun to diminish, as a result of the perceptual and articulatory bias against this peak; once this has happened, generation 2 can move the unimodal category into the vacating region on its left. The unimodal category now drags the (ever growing) right peak of the bimodal category along towards the centre of the continuum, until the system consists of two unimodal categories.

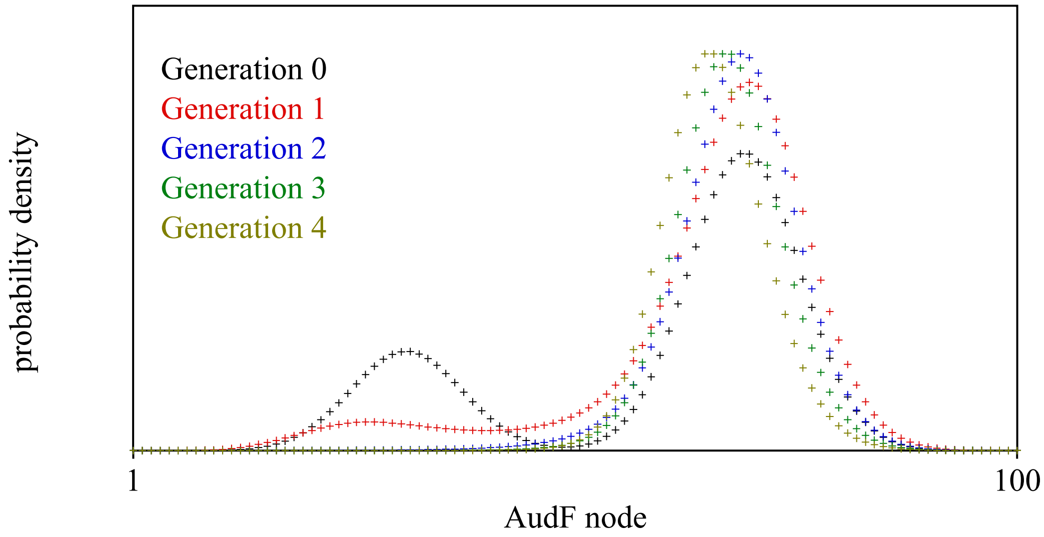


Figure 25. *The gradual disappearance of a discontiguous category.*

The inventory from Figure 23, in which both peaks of the bimodal category have the same probability, evolves as in Figure 26. By chance, now it is the right peak that disappears eventually: apparently, auditory values to the left of the centre appear slightly more often in the input of generation 1 due to random factors, namely because they are selected more often during perceptual learning and/or because of the transmission noise. Once a bias against one of the peaks has arisen, this bias is reinforced by the articulatory effect, so that this smaller peak vanishes eventually.

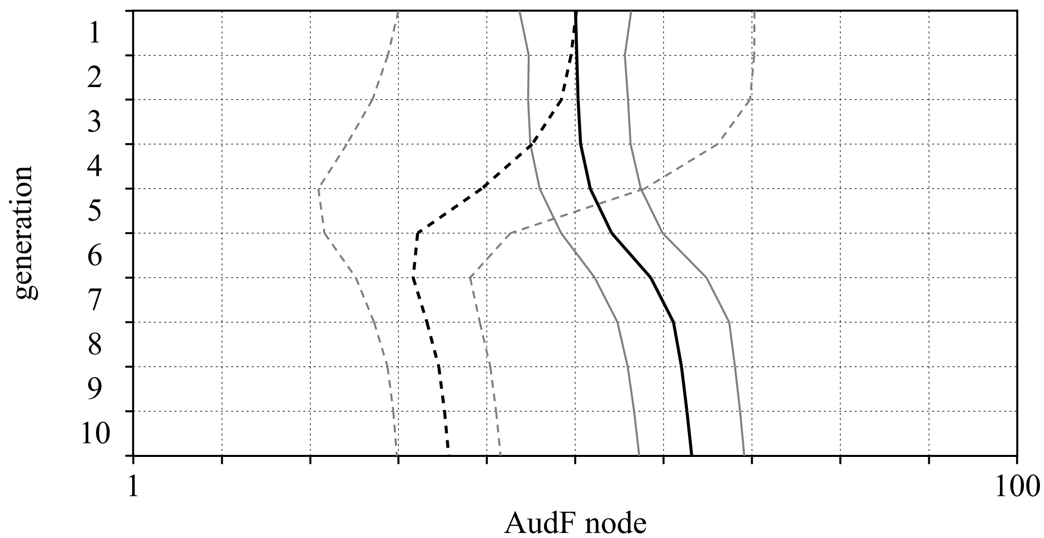


Figure 26. *The evolution of the inventory from Figure 23 (outstar learning).*

Figure 26 makes clear that a symmetric bimodal distribution evolves into an optimally dispersed, unimodal distribution as well, so the demise of the discontiguous category in B&H's simulations was not due to the fact that one of the peaks was preferred from the very beginning.

6.2 Inoutstar learning

6.2a Two-category inventories (with contiguous categories)

In general, the cue connections are weaker in an inoutstar network than they are in an outstar network (remember Figures 12 and 13, p. 26–27), so if we were to maintain the articulatory settings from the outstar simulations, the categories in the final state would come to lie unrealistically close together. I adjust them to values that result in an equilibrium similar to that from the outstar simulations: the degree of inhibition is taken to be 0.5 instead of 1.5, and the weight of the central connection becomes -0.1 instead of -0.25. All other parameter settings in these simulations are the same as in the previous section.

With inoutstar learning, a standard initial distribution remains stable over the generations. The output probability curves of generation 1 are bell-shaped and appear to be symmetric:

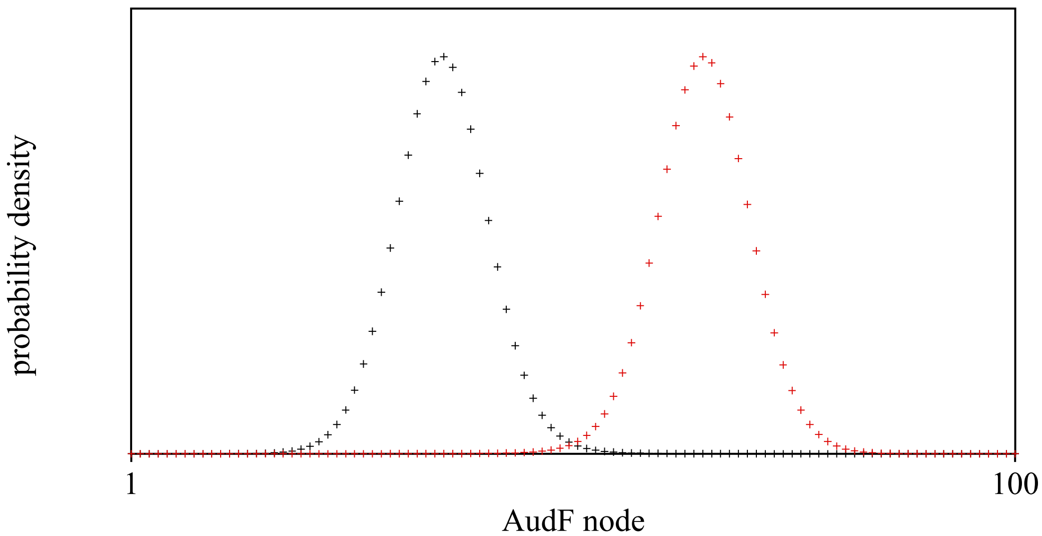


Figure 27. *Output probabilies of generation 1 (standard distribution, inoutstar learning)*.

Other types of initial distributions evolve into optimally dispersed systems in inoutstar networks as well. However, they evolve quite differently than they do in an outstar network: because the cue connection weights in an inoutstar network mirror the input distribution more closely than they do in an outstar network (§5.2), the in- and outputs of a generation differ less than in an outstar network, so that changes proceed slower and more gradually. An exaggerated contrast takes some 12 generations to stabilize, whereas three sufficed in an outstar network:

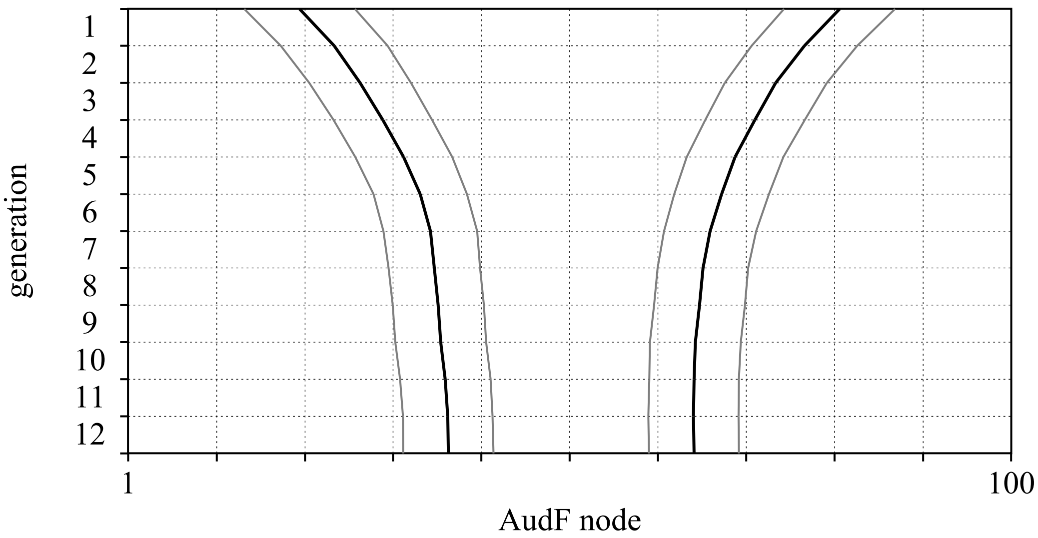


Figure 28. *The evolution of an exaggerated initial distribution (inoutstar learning).*

A confusing opposition needs more than 20 generations to be resolved, instead of five in an outstar network:

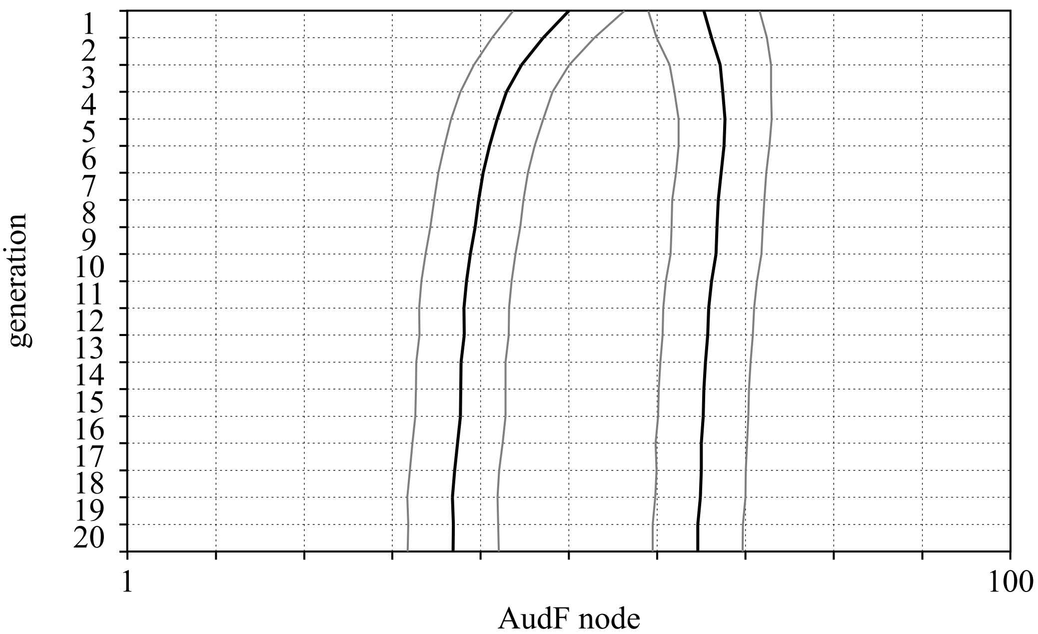


Figure 29. *The evolution of a confusing and skewed initial distribution (inoutstar learning).*

Nevertheless, irrespective of the initial distribution, two-category inventories evolve into stable systems in inoutstar networks too.

6.2b Three-category inventories: Polish sibilants

In simulations with inoutstar networks, a larger number of categories required a shallower articulatory curve (cf. §6.1b, particularly Figure 19, p. 33). In an inoutstar network, by contrast, satisfactory results are achieved with the same parabola used in the two-category simulations from §6.2a:

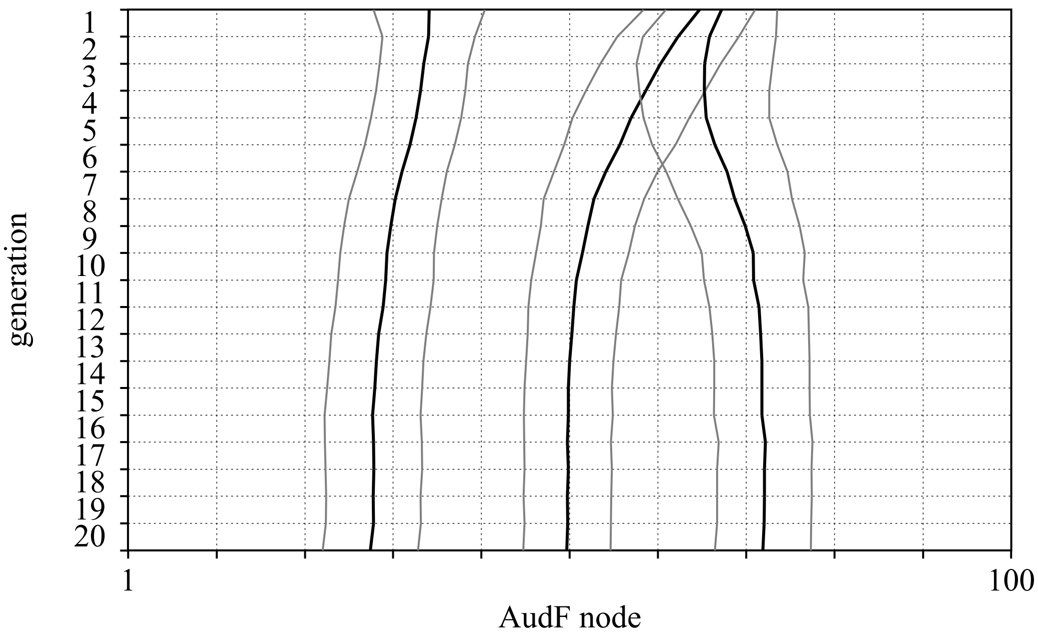


Figure 30. *The evolution of medieval Polish (inoutstar learning).*

This simulation of medieval Polish predicts a smoothly proceeding evolution with a final state in which the categories lie only slightly closer together than in Figure 20 (p. 34): category /A/ lies at 27.4% of the auditory continuum, /B/ at 49.7% and /C/ at 71.9%. The difference between the average produced nodes of /B/ and /C/ increases from the very first generation on, but we do not see the marked enhancement in the first generation of learners predicted in B&H's simulations: the articulatory effect pushes the categories into a less effortful region. The inventory again occupies a larger part of the auditory continuum than a two-category inventory: the distance between the average produced nodes of the extreme categories spans 44.5 and 27.8% of the continuum, respectively.

6.2c One-category inventories

Considering the relative slowness with which asymmetric distributions approach an equilibrium in inoutstar networks, any skewness within a one-category inventory will probably not have been resolved within a single generation, as in B&H, nor within two generations, as in an outstar network (cf. §6.1c). In fact, even when the peak in the input distribution lies at 20% of the continuum (instead of 10% in §6.1c), the system takes ca. 30 generations to become optimally dispersed:

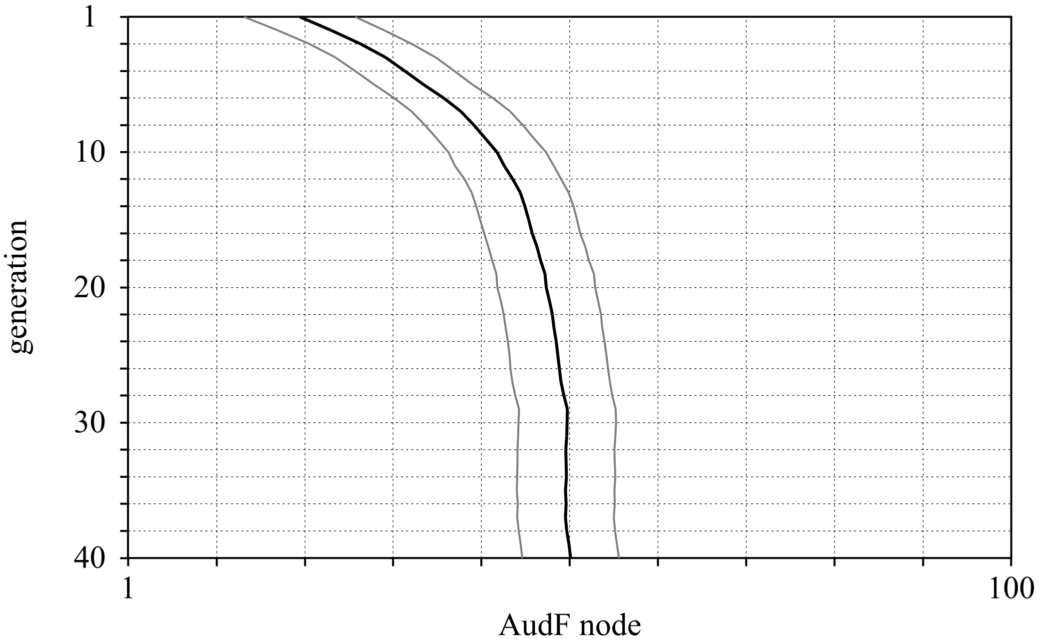


Figure 31. *The evolution of a skewed one-category inventory (inoutstar learning).*

Nevertheless, the inventory eventually reaches the stable state we expect, in which the average produced node lies halfway the auditory continuum.

6.2d Two-category inventories (with one non-contiguous category)

A bimodally distributed category becomes unimodal in an inoutstar network as well. Figure 32 shows the evolution of the symmetric distribution from Figure 23 (p. 37), in which both peaks have the same probability.

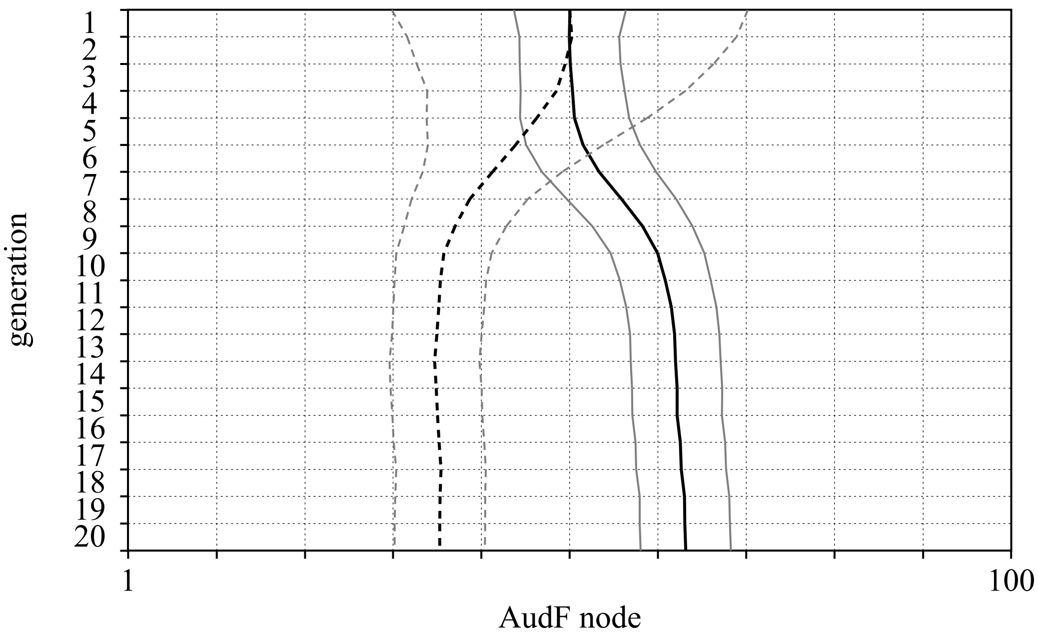


Figure 32. *The evolution of an inventory with one bimodally distributed category (inoutstar learning)*.

One peak of the bimodal category disappears (by chance again the right one), and the standard deviation of this category normalizes within some ten generations. Since this 'unbiased' distribution becomes unimodal, I assume that an asymmetric, 'biased' distribution (Figure 22, p. 37) will too; I do not provide simulations of this here.

6.3 The evolution of unbounded auditory contrast

The script offers the possibility to run simulations without an Articulatory Form, in order to investigate the contribution of the articulatory effect to the results of the simulations. The results of such simulations are presented in this section. With the Articulatory Form switched off, the output activity of an AudF node given a category is equal to the cue connection weight between these two nodes: thus, we may expect considerable differences between outstar and inoutstar networks.

6.3a Outstar learning

In a two-category inventory with a standard initial distribution, the average produced nodes immediately draw towards the periphery, and the standard deviations increase; after that, the system remains stable:

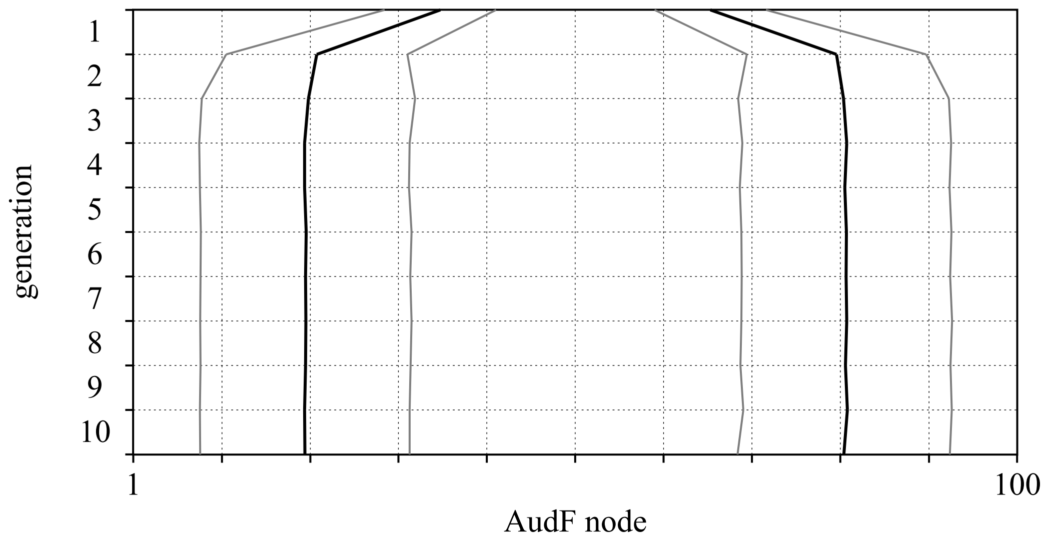


Figure 33. *The evolution of a two-category inventory without the articulatory effect (outstar learning)*.

Upon further investigation, we can see that this system has some unrealistic properties. Figure 33, like similar figures from previous sections, was drawn with a script that draws lines connecting the average produced nodes of every category, and this value ± 1 standard deviation. In the networks from previous sections, this made sense because the output probability curves were bell-shaped; however, they are not in this network, so Figure 33 is highly deceptive. In fact, Figure 34 shows that with the articulatory effect left out, the output probabilities show minimal overlap and are not restricted towards the periphery anymore (remember that outstar learning is sensitive to the degree to which an output is predictable from the input, not to the frequency of this input):

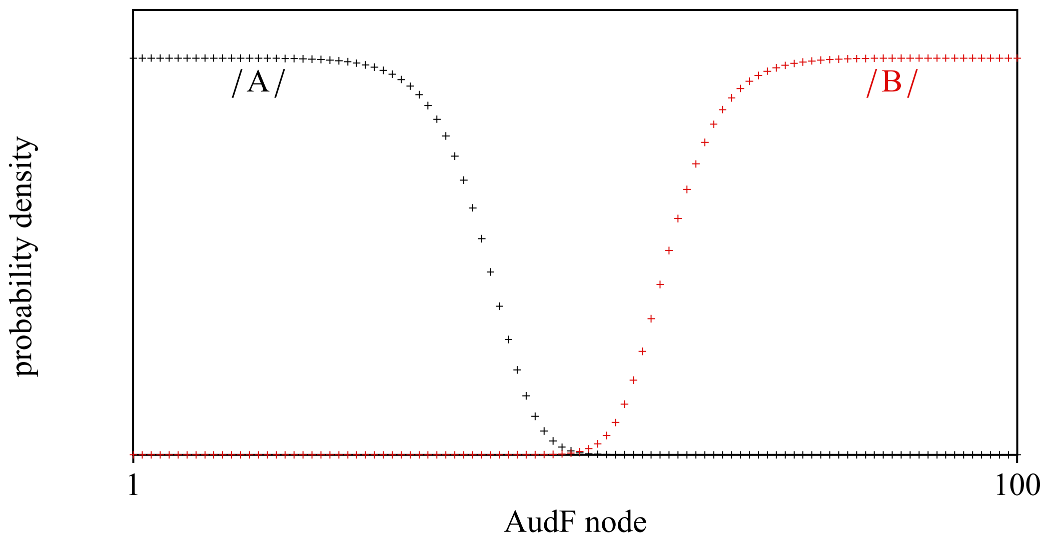


Figure 34. *The output probabilities of generation 1 from Figure 33.*

The evolution of unbounded medieval Polish is shown in Figure 35: now we can answer the question whether an even shallower articulatory curve would have yielded better results. Even without an Articulatory Form, the upper two categories do not move in opposite directions in the first generation, while the lowest category immediately shifts down: there would be enough room for the middle category to drift away further from the highest category (but see also §6.6). Nevertheless, the phonetic enhancement of the contrast between the right two categories is larger than in the simulations from §6.1b, but this contrast only seriously increases after the first generation.

As in the previous simulation, the unbounded inventory eventually comes to occupy the entire available auditory space.

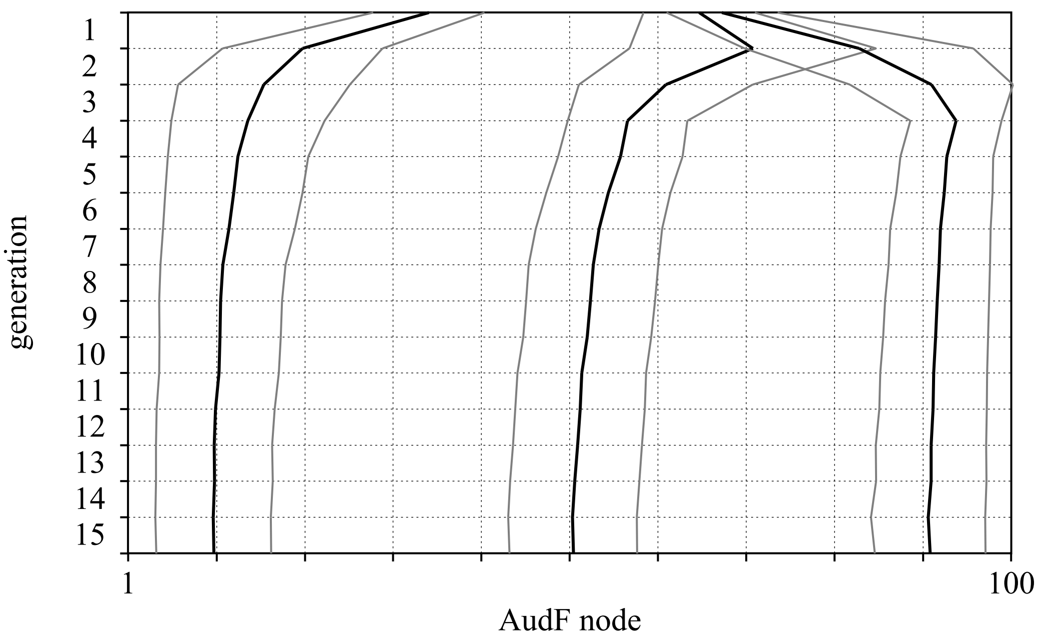


Figure 35. *The evolution of medieval Polish without the articulatory effect (outstar learning).*

In the simulations of the evolution of inventories with one asymetrically distributed bimodal category (§6.1d, 6.2d), the disappearance of one of the peaks of this category is caused by an interaction of perceptual and articulatory factors: the cue connections to the most frequent peak are stronger, so this peak is chosen more frequently in production; as a result, this peak becomes larger and pushes the unimodal category away, which in turn pushes the smaller peak of the bimodal category towards the periphery of the continuum. There, it is chosen less often because of the articulatory effect.

In the first place, then, it is the prototype effect that favours the larger peak; we would thus expect bimodal categories to become unimodal in the absence of the articulatory effect as well. I will show that they do in inventories with a symmetrically distributed bimodal category, where the prototype effect in itself will not favour one of the peaks. In the previous simulations with an Articulatory Form, it was the transmission noise and/or a slightly unbalanced input that caused one of the peaks to overtake the other, i.e. this happened by coincidence; once it had happened, the prototype effect did start to play a role, and was reinforced by the articulatory effect. In an unbounded inventory, where the articulatory effect does not add to the process, the evolution proceeds rather slowly compared to Figure 26 (p. 39), but two unimodal categories emerge:

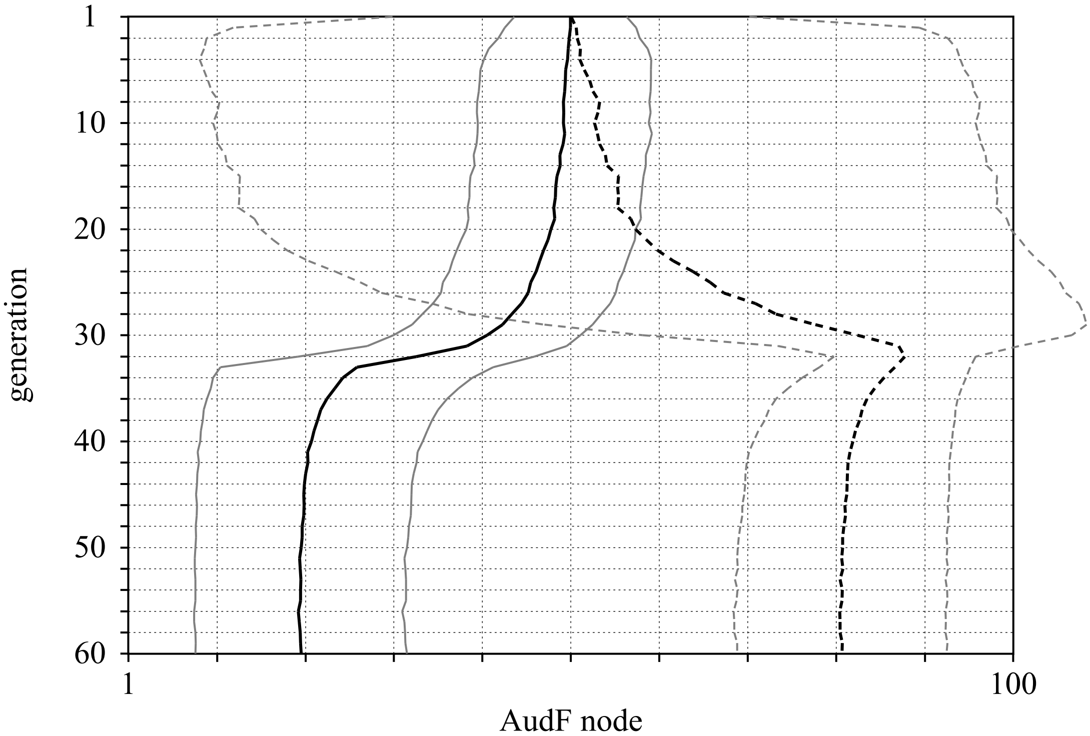


Figure 36. *The evolution of an inventory with one symmetrically distributed bimodal category, without the articulatory effect (outstar learning): this category becomes unimodal.*

A one-category system, irrespective of the initial distribution, comes to occupy the entire continuum (Figure 37). Since no between-category contrast needs to be maintained and there are no articulatory restrictions on the system, all auditory values have the same probability in production (Figure 38):

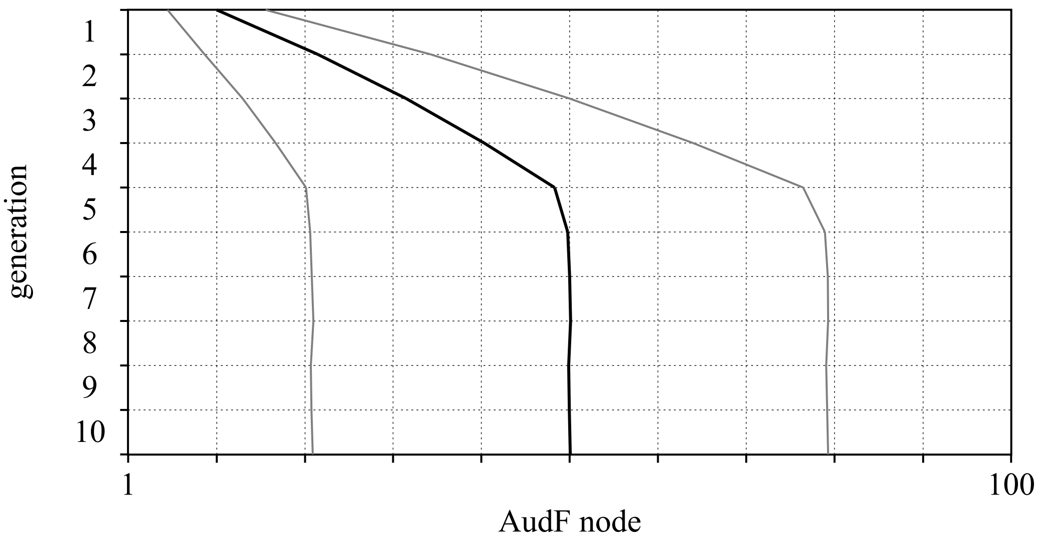


Figure 37. *The evolution of a skewed one-category inventory without the articulatory effect (outstar learning).*

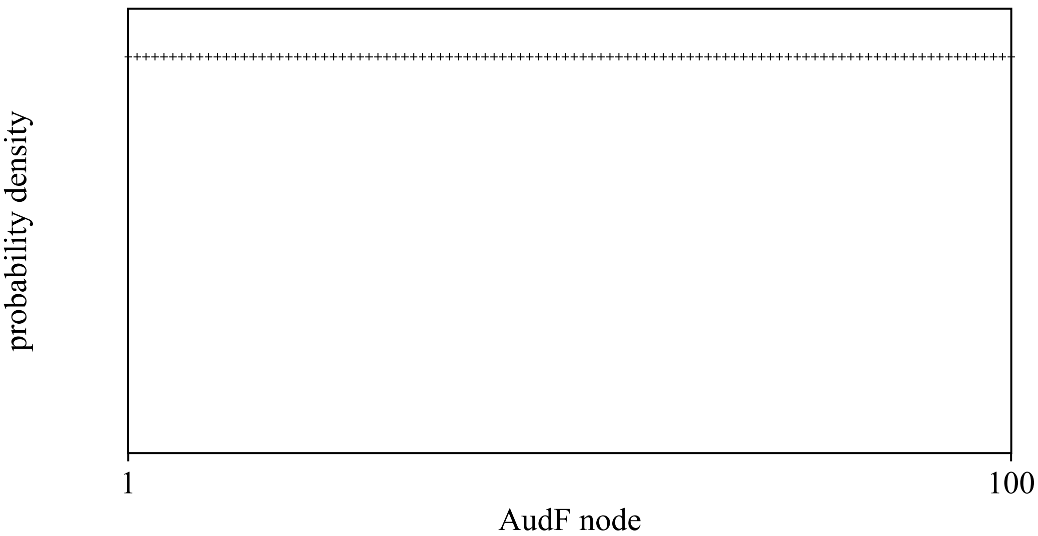


Figure 38. *The output probabilities of generation 10 from Figure 37.*

6.3b Inoutstar learning

We have observed that inoutstar learning is more faithful to the input distribution than outstar learning, which is sensitive to predictability rather than frequency; this faithfulness is seen even in the absence of articulatory inhibition.

In a two-category inventory, the distributions do move away from each other, but only very slowly (Figure 39). They seem to stabilize after generation 19 (although this cannot be concluded from the figure with absolute certainty), probably because the cue connections hardly show any overlap anymore. In sharp contrast with the unbounded outstar networks, the output probability curves in unbounded inoutstar networks remain bell-shaped, even after tens of generations (Figure 40); thus, in this subsection we do not observe the dislodgement of the output probability curves seen in §6.3a.

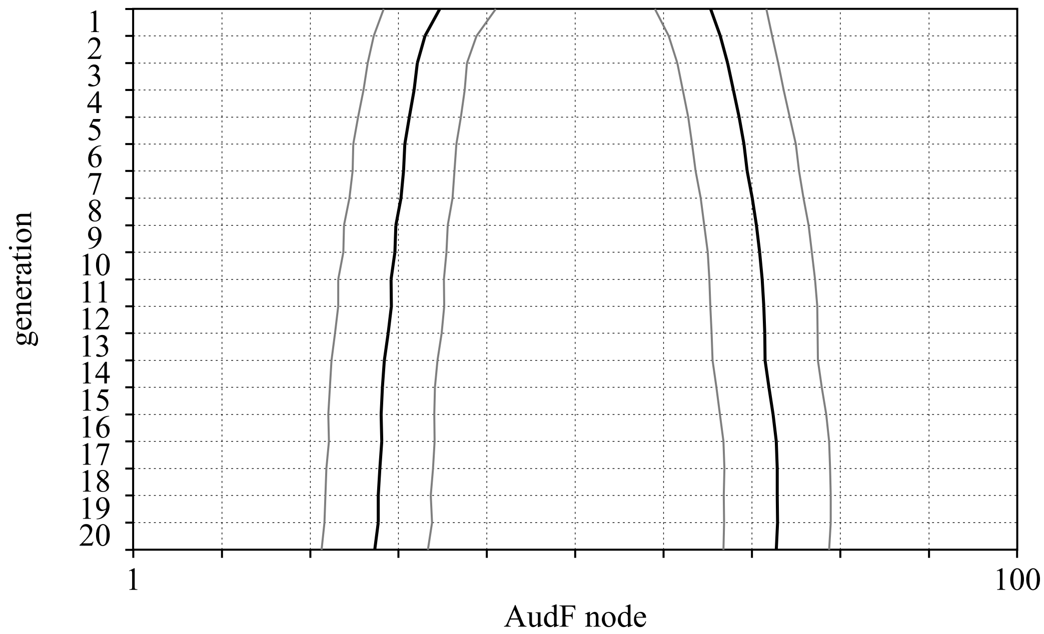


Figure 39. *The evolution of a two-category inventory without the articulatory effect (inoutstar learning)*.

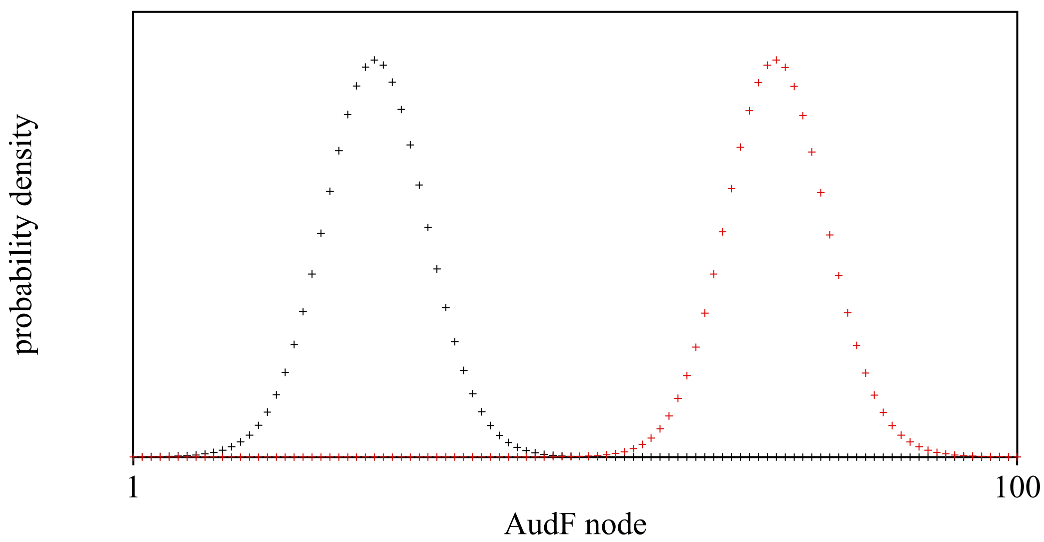


Figure 40. *The output probabilities of generation 20 from Figure 39.*

The output distribution of a skewed one-category inventory does move away from the edge of the continuum towards the centre, but at a lower pace than the categories in a larger system do.

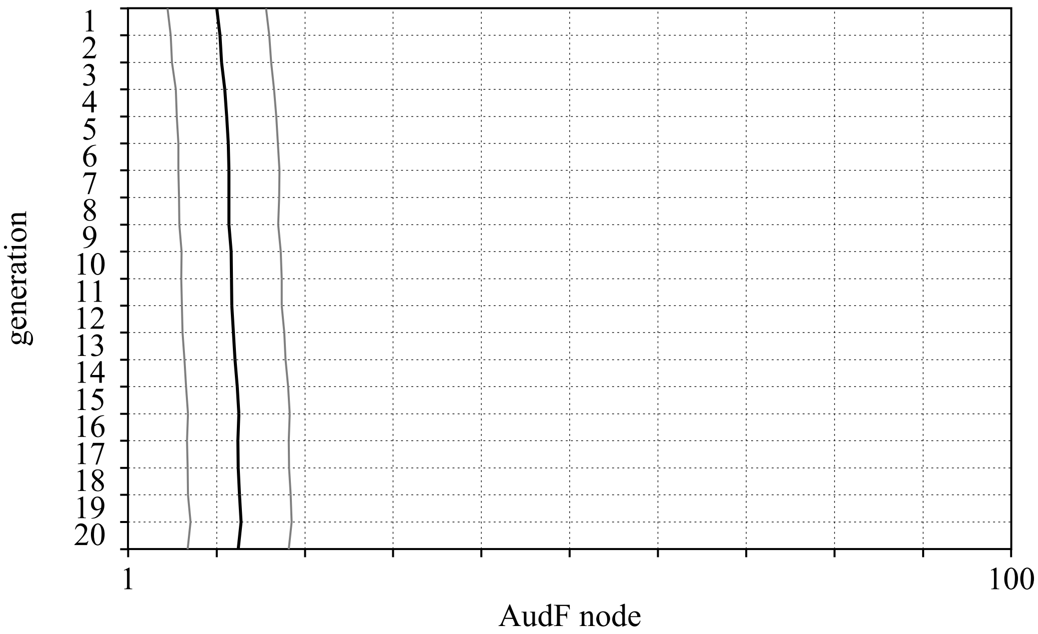


Figure 41. *The evolution of a skewed one-category inventory without the articulatory effect (inoutstar learning)*.

The motion in this inventory seems to halt around generation 15. In a one-category inventory there is no need to avoid overlap in the cue connections, because there cannot be any overlap (after all, there are no competing categories); since the inoutstar learning rule yields cue connection weights that reflect the shape of the input distribution fairly faithfully, we might not even expect the category to move at all in the absence of the articulatory effect. However, it does in Figure 41, which is probably a consequence of the fact that in the initial distribution, the leftmost node has a relatively high probability of .346 before transmission noise; whereas this transmission noise can in principle scatter a perceived auditory value to either side of the intended value, the script forces the perceived value within the range of AudF nodes of the network, so it cannot be lower than 1 here. By exception, the average perceived node when node 1 is produced is thus higher than 1. This results in a slight rightward movement of the distribution, which slows down considerably when node 1 apparently occurs in the input so infrequently that it does not exert much influence anymore. Nevertheless, because of this 'range effect', the distribution is expected to shift to a central value very slowly.

In an unbounded inoutstar network, an inventory with a bimodally distributed category is also unstable without the articulatory effect: once one of the peaks has become more frequent than the other, this peak will overtake the less frequent one. In this respect, an inoutstar network is very fast, much more so than an outstar network: because it retains the shape of the output probability curves to a larger extent, it behaves more like the networks with an Articulatory Form from §6.2.

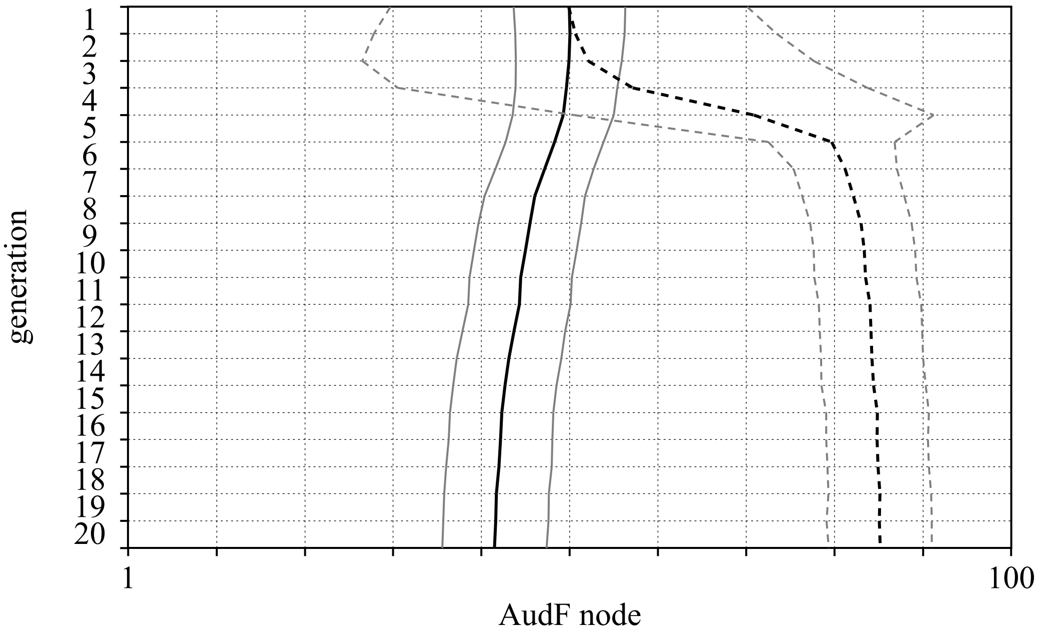


Figure 42. *The evolution of an inventory with one bimodal category without the articulatory effect (inoutstar learning).*

As in the previous simulation of Figure 41, this inventory seems to reach an asymmetric final state; any change proceeds much slower from the moment that the cue connections hardly overlap anymore. As in Figure 39 (p. 49), this apparently happens when the average produced nodes are spaced ca. 45 nodes apart. However, because of the 'range effect' mentioned before, category /B/ might eventually towards the centre of the continuum, pushing /A/ away.

The simulations in this section show that the prototype effect causes any system – standard, skewed and confusing, with contiguous and non-contiguous categories – in any type of network – outstar and inoutstar – to evolve into a state with minimal confusability. Outstar networks evolve into stable states with symmetric output distributions that are not bell-shaped anymore, but unrestricted towards the periphery;

inoutstar networks also evolve into stable states, but require more time to do so, and remain more faithful to the initial output probability curves. The apparently stable states of unbounded inoutstar networks are not necessarily symmetric. This may seem like an undesirable property, but it should be kept in mind that the simulations presented here do not replicate real-world phenomena. They are intended to investigate the evolution of inventories without articulatory factors, and indeed do just that: they all yield stable inventories in which categories are minimally confusable. In addition, they also make clear that the articulatory effect is needed to make accurate predictions about the auditory correlates of phonological categories, and thus provide support for a formalization in which these effects interact.

6.4 Outstar vs. inoutstar networks

The previous sections have shown that any phonological system is predicted to evolve into an optimally dispersed inventory, both in outstar and inoutstar networks, and that the prototype effect and the articulatory effect are both indispensable to achieve this. Optimal auditory dispersion emerges much slower in inoutstar networks than in outstar networks; however, there is only meager and indirect evidence about how quickly an exaggerated or confusing contrast would be resolved in natural languages (remember the Polish sibilant system, and cf. §6.5). Advantages of inoutstar learning are its bidirectionality, and the fact that it has also proven capable of explaining another phenomenon in the phonology-phonetics interface, viz. the perceptual magnet effect (Boersma, Benders & Seinhorst fc.).

A major difference between outstar and inoutstar networks are the cue connection weights (cf. Figures 12 and 13, p. 26–27): because of this, both kinds of networks make different predictions about which token is the prototype of a category (remember from §5.2 that the prototype has the strongest cue connection to the category node). Figure 43 shows the cue connection weights of the virtual learner from Figure 15 (p. 29; outstar learning):

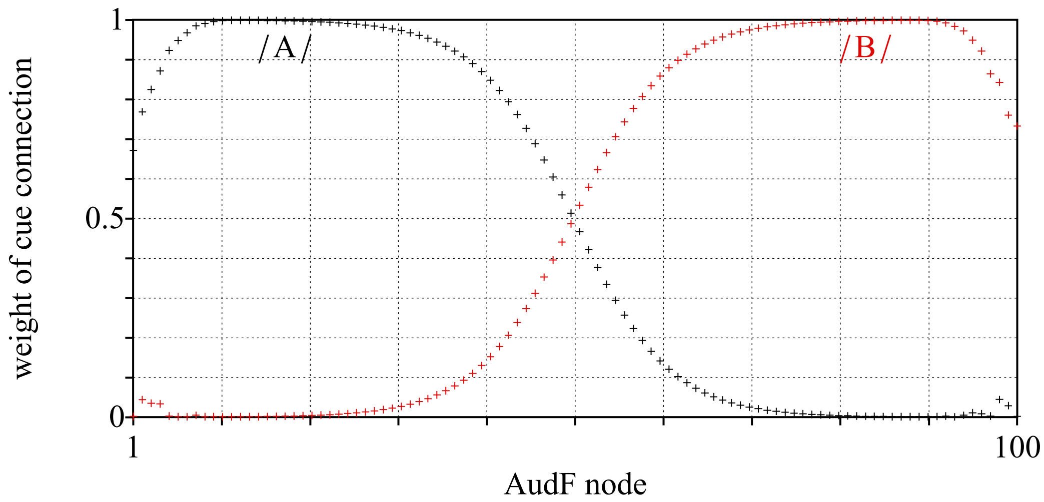


Figure 43. *Cue connection weights of the virtual learner from Figure 15 (outstar learning).*

In an outstar network, there is an array of tokens with very strong cue connections, which we could perhaps consider a prototypical region. This region is much more peripheral than the region of maximal activity in production: for category /A/, the peak in production lies at 37.4% of the continuum, the prototype at 12.1%. They are thus spaced 25.3% apart.

In an inoutstar network, by contrast, there is no such prototypical region: here we see normally distributed cue connection weight curves. Figure 44 shows the cue connection weights of the virtual learner from Figure 27 (p. 40; inoutstar learning):

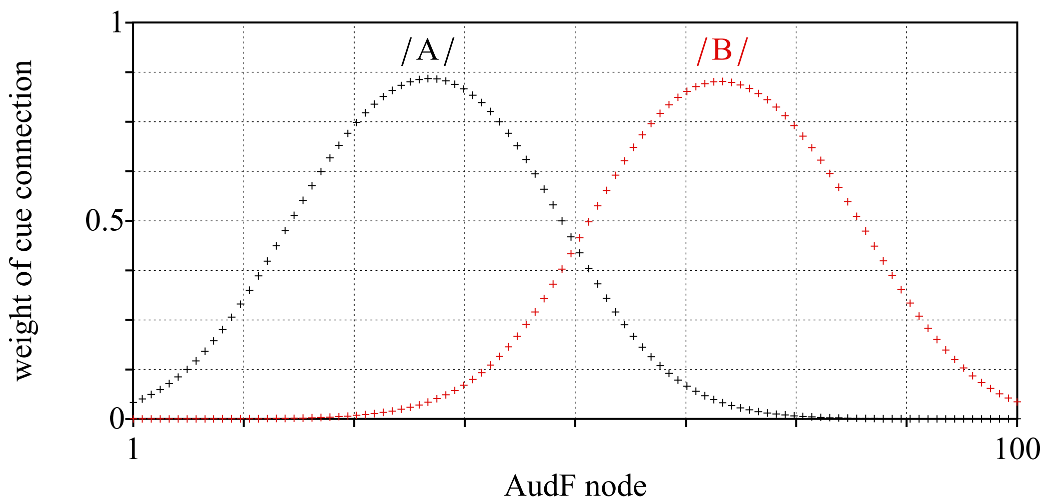


Figure 44. *Cue connection weights of the virtual learner from Figure 27 (inoutstar learning).*

In this network, the prototype lies close to the most active node in production: in production, the peak of category /A/ lies at 35.4% of the continuum, and the prototype at 33.3%, i.e. they are 2.1% apart.

Although the two types of network make different predictions about the location of the prototype, they both replicate Johnson, Flemming & Wright's (1993) findings that listeners prefer more extreme auditory values in perception than they actually produce on average themselves (modelled in OT by Boersma 2006). It cannot be determined which prediction is more realistic: the auditory space that Johnson, Flemming & Wright investigated was two-dimensional and contained eleven categories, so any comparison to the predictions made here concerning a two-category inventory in a unidimensional space would be to no avail.

6.5 The auditory dispersion of the Dutch sibilant inventory

B&H (footnote 14, p. 254) speculate that a bimodal distribution may emerge (and eventually disappear again) in the Dutch sibilant system, where a "relatively new" alveolopalatal /ɕ/ [ɕ] has been introduced into an inventory containing the flat laminal /s/ [s̺]. Since both these sounds have central values of the spectral mean, B&H hypothesize that transitory allophony may occur, i.e. in some varieties of Dutch the spectral mean of [ɕ] may come to lie above that of [s̺], and below it in other varieties.

Upon reading this speculation some three years ago, I have at the same time been fascinated and puzzled by it, because – in spite of their similar spectral means – the Dutch sibilants do not seem to be confusable. [ɕ] is not only the sound we find in loanwords like *show* and *chic*, but also the realization of an underlying sequence |s+j|, e.g. in *tasje* |tɑs+jə| [tɑɕə] "bag-DIMINUTIVE". Here it contrasts unproblematically with [s̺]: Dutch also has the word *tassen* |tɑs+ə| [tɑs̺ə] "bag-PLURAL". Thus, an inventory with poorly dispersed spectral centres of gravity functions perfectly well in practice.

B&H label the phoneme /ɕ/ as "relatively new" and "somewhat marginal", undoubtedly because of its occurrence in loanwords. The *sound* [ɕ], on the other hand, is certainly not marginal, and may not be new either: in any case, sequences of underlying |s+j| have been present in Dutch for centuries. For instance, P. C. Hooft wrote in 1617 *'t Meysje was moy, ick haddet altijt wel moghen zien* 'The girl was pretty, I would have wanted to see it all the time' (Warenar, line 968). Obviously, it is unknown how such a sequence was pronounced then, but the options that probably strike an optimal balance between perceptual distinctiveness and articulatory ease are homorganic sounds, such as [ʃ] or [ç]; other (heterorganic) possibilities are [s̠j] or [s̠ʲ].[5]

The *phoneme* /ç/ was introduced into Dutch through a process of borrowing in the 19th century (Van der Sijs 2002: 214), which should have caused a shift in the Dutch sibilant system. Perhaps this shift is only occurring right now, sped up by a new influx of loanwords with /ç/ from English in the late 20th century, but this would mean that the realizations of underlying |s+j| and /ç/ have coalesced very quickly.

In any case, there was no transitory allophony in the sample of speakers in Seinhorst & Ooijevaar (2011), all ten of whom had higher spectral means in [s̠] than in [ç]. These categories behave as if they are the only one in the inventory, i.e. they have central values of the spectral mean ([ç]: 27.26 Erb; [s̠]: 29.39 Erb) and a large standard deviation ([ç]: 0.96 Erb; [s̠]: 1.10 Erb) compared to a control group of two French speakers (basic statistics: mean: [ʃ]: 26.20 Erb, [s]: 31.45 Erb; standard deviation: [ʃ]: 0.39 Erb, [s]: 0.41 Erb):

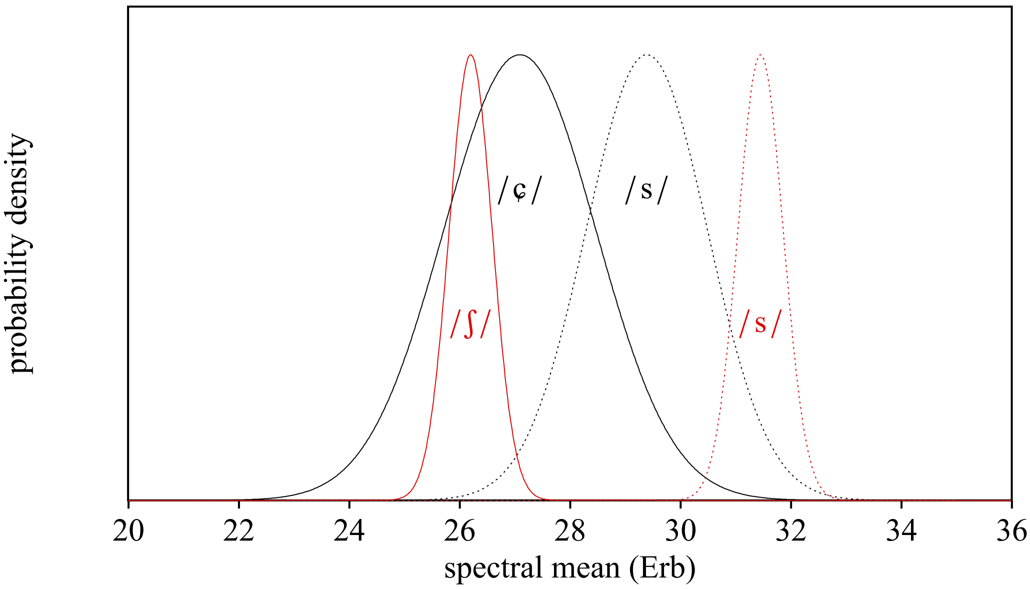


Figure 45. *The distributions of the spectral means of Dutch and French sibilants (Dutch: black line, French: red line)*.

[5] The pronunciation [s̠i] for |s+jə| is also conceivable, and is in fact quite common in colloquial contemporary Dutch: *wasje* '(small) load of laundry' can be pronounced [ʋɑs̠i] instead of [ʋɑçə] (and may be written accordingly). In this case, Dutch would indeed have had only one sibilant in Hooft's days.

It remains to be seen whether the Dutch sibilant inventory is diachronically stable: if the spectral mean is decisive in the auditory dispersion of sibilant systems, the Dutch system is expected to come to resemble the French inventory eventually (outstar and inoutstar simulations make different predictions about the time stretch in which this happens). If the spectral means in the Dutch sibilant inventory indeed become optimally dispersed, a problematic issue remains, because it would imply that the introduction of /ɕ/ caused the change. This would entail that the pronunciations of |s + j| and /ɕ/ have merged quickly, while at the same time the spectral means of these categories have not become optimally dispersed. It is unclear to me why one sound change should have proceeded at a much higher pace than another. However, all this is somewhat speculative.

6.6 The parameters of the network, and the activity-to-probability rule

The simulation script asks the user to enter a number of parameters (besides the locations of the peaks in the probability distributions). These are: the number of learning steps (§5.1); the transmission noise (§5.2); the number of AudF nodes (§5.1); the degree of articulatory inhibition (§5.1); the weight of the central articulatory connection (§5.1); the temperature (§5.2); the values of instar and outstar (§4.3, already explored in §6.1–6.4); the presence or absence of an Articulatory Form (§5.1, already explored in §6.3); the learning rate (§4.3); and the activation clipping rule (§5.2). Additionally, a number of preset parameters is used in the script. Apart from the network settings (e.g. minimal/maximal values of the activity of unclamped nodes, minimal/maximal values of a connection weight), these are the standard deviation of the category (§5.2) and the standard deviation of the excitation spreading along the basilar membrane (§5.2).

The network needs a sufficient number of learning steps to exhibit realistic behaviour. If the number of learning steps is taken very low, say 1000, the inventory becomes highly unstable: the average produced nodes of a category vary greatly between generations, as do the standard deviations, and even merger may occur.

Increasing the transmission noise results in larger within-category variation and more fluctuation in the evolution of the category. Also, it predicts a larger between-category distance.

Not surprisingly, decreasing the degree of articulatory inhibition and/or the weight of the articulatory central connection results in a greater between-category distance.

Additionally, the standard deviations are restricted to a lesser degree, allowing them to become larger; and the output probabilities are higher at the periphery of the auditory continuum, which means that the probability curves are not symmetric.

The temperature indicates the degree of evaluation noise, so a lower temperature should result in lower within-category variation. Indeed, decreasing the temperature by a factor 10 increases the probability of the node with the highest activity by a factor e^{10} and results in much smaller standard deviations. As a consequence, inventories are diachronically more stable (i.e. the average produced nodes fluctuate less) when the temperature is low.

The standard deviation of a category in the optimally dispersed state is independent of that in the initial distribution: rather, it is determined by the articulatory settings, the transmission noise and the temperature, and any initial standard deviation will stabilize at this value (this does not mean that all categories have the same standard deviation in the final state; cf. Figure 20, p. 34).

The reader may object that the standard deviation of the activity spreading along the AudF layer (Figure 11, p. 25), viz. 10% of the number of AudF nodes, is unrealistically large, and (s)he would be right (cf. Moore & Glasberg 1983). This value was chosen because it yielded symmetric output probability curves for the peripheral categories; smaller standard deviations predict curves that are shallower at the edge of the continuum. This may very well be realistic, because no auditory contrast needs to be maintained in these areas. Networks with a smaller standard deviation of this activity spreading learn somewhat slower, because fewer nodes are activated in each learning step, but optimal auditory dispersion does emerge; and because a category is associated with a smaller range of AudF nodes, overlapping (i.e. confusing) distributions are resolved more quickly. Remember that even in the simulation of the Polish sibilant inventory without an Articulatory Form (Figure 35, p. 46), the upper two categories drew in the same direction. If the standard deviation of the activity spreading is reduced from 10% to 3.33% of the number of AudF nodes (and the number of learning steps is increased from 100,000 to 250,000 as a strategy to compensate for this), these categories do move apart:

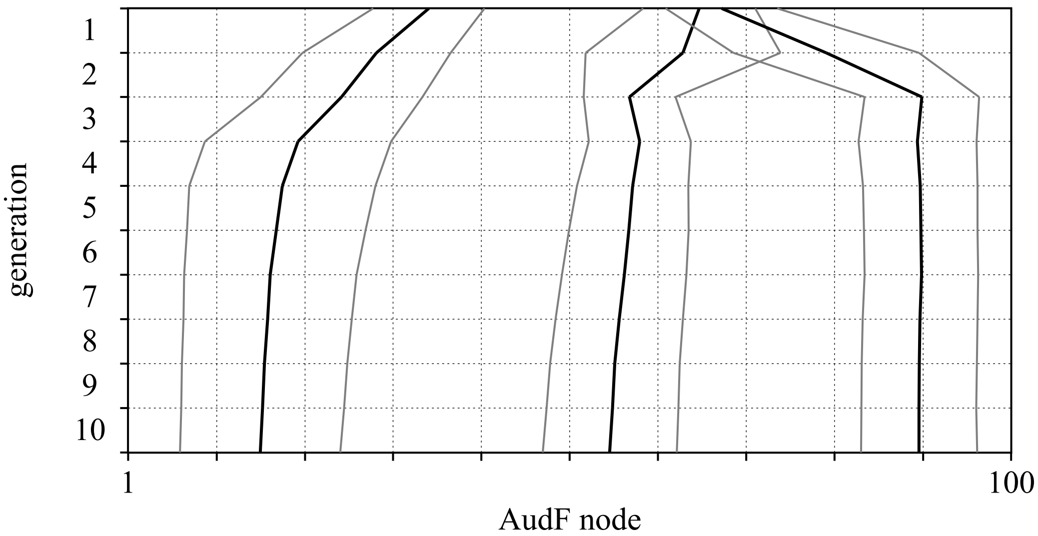


Figure 46. *The evolution of medieval Polish without the articulatory effect (outstar learning, standard deviation of the activity spreading = 3.33 nodes).*

Decreasing the learning rate requires a longer learning phase (i.e. more input tokens), but predicts more diachronic stability. Particularly inoutstar networks are sensitive to this parameter setting: because the cue connection weights tend to be fairly small, a too high learning rate causes the weights to fluctuate depending on what category appeared in the input last.

The activation clipping rule influences the shape of the output probability curves, the within-category variation and the locations of the categories in the final state: however, optimal auditory dispersion emerges independently of the clipping rule.

So far, all networks in this thesis have used an exponential activity-to-probability rule ($p = e^{a/T}$); remember from §5.2 that activities can also be interpreted directly as relative probabilities. This obviates the need for a temperature parameter. Figure 47 shows the evolution of auditory dispersion in an inoutstar network (with 50 AudF nodes) where the activity of a node is equal to its output probability, i.e. with a linear activity-to-probability rule: this network is fed a standard distribution, i.e. the peaks in the input lie at 35 and 65% of the continuum. The parameter settings in this simulation are: 250,000 learning steps; articulatory inhibition 0.05; weight of the central articulatory connection -0.05; linear activation clipping between 0 and 1; transmission noise 0.025; standard deviation of the activation spreading around the perceived AudF node 0.025.

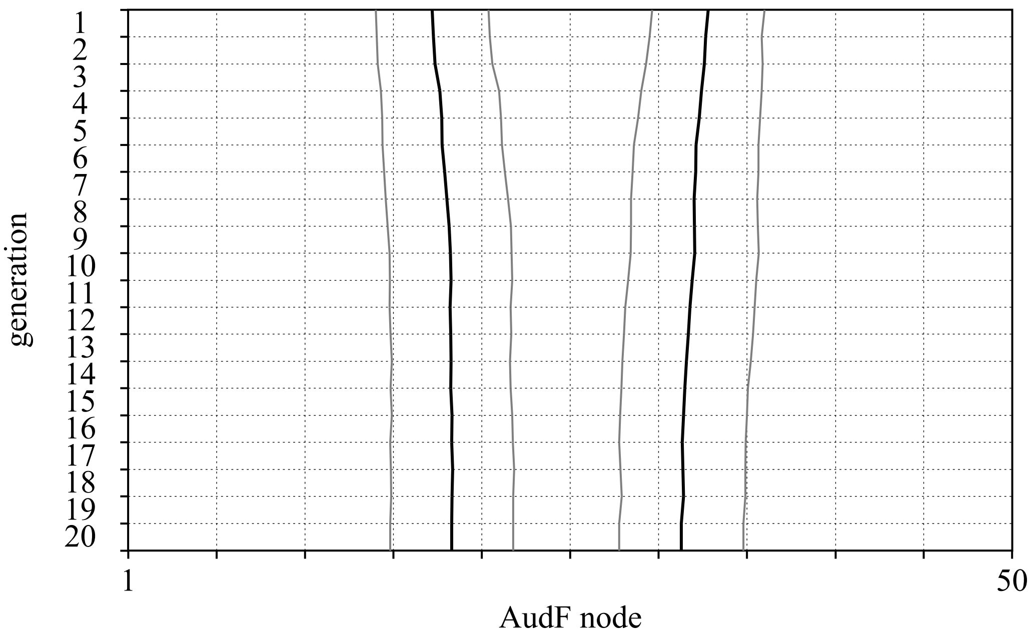


Figure 47. *The evolution of auditory dispersion in an inoutstar network with a linear activity-to-probability rule.*

We see that the network reaches an equilibrium, but any change proceeds very slowly because this network requires very weak articulatory inhibition. This slow pace becomes even more apparent in the evolution of an inventory where the peaks in the initial distribution lie at 50 and 65% of the continuum, i.e. the skewed and confusing distribution from §6.1a and 6.2a), shown in Figure 48. After 50 generations, the categories lie at ca. 42 and 67% of the auditory continuum, the positions that they had already reached after five generations in an inoutstar network with an exponential activity-to-probability rule (which, in turn, predicts slower changes than an outstar network with an exponential activity-to-probability rule – cf. Figures 29 (p. 41) and 18 (p. 32) respectively):

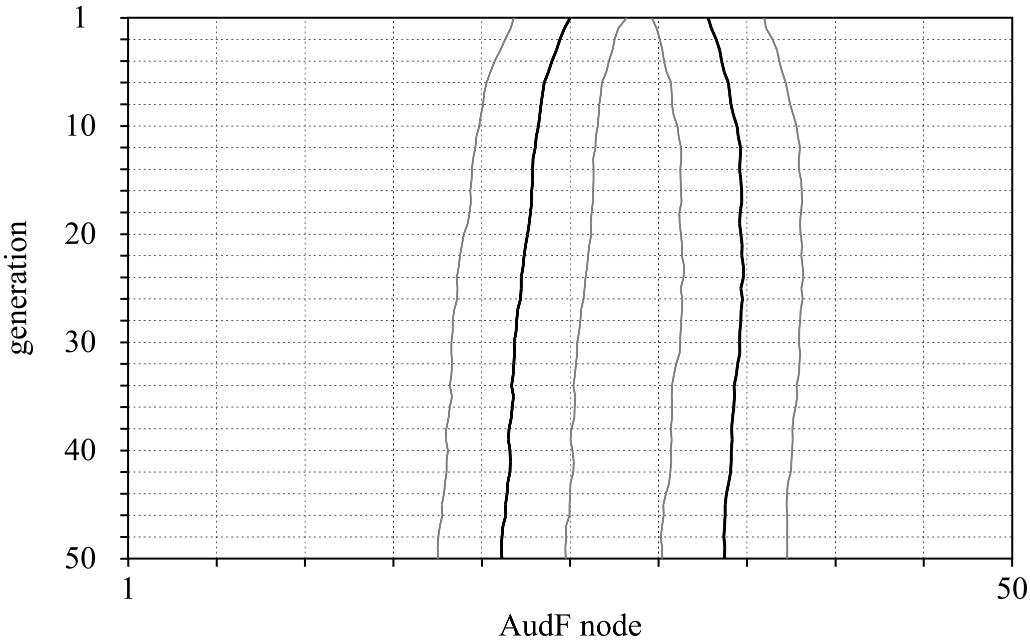


Figure 48. *The evolution of a confusing and skewed initial distribution in a network with a linear activity-to-probability rule (inoutstar learning).*

Optimal auditory dispersion is predicted to emerge in networks with a linear activity-to-probability rule, but considerably slower than in networks with an exponential activity-to-probability rule.

7 The learnability of phoneme inventories, and a possible quantification strategy

This paragraph discusses the learnability of phoneme inventories. As such, I consider the degree to which such inventories (or, in fact, linguistic systems of whichever nature: morphosyntactic, semantic etc.) are transmitted faithfully from parent to child. Christiansen & Chater (2008) argue that the inductive biases in the human brain have shaped languages in a process of iterated learning: systems that do not comply with our learning preferences are difficult to learn and will be diachronically unstable. Similarly, a phoneme inventory will change if it does not properly balance our perceptual and articulatory biases, i.e. if the prototype effect and the articulatory effort are of unequal size. This hypothesis is corroborated by observations about real languages: poorly dispersed inventories are not stable (remember the Polish sibilant system), which is why we do not find them, or perhaps only rarely; and languages have a preference for optimally dispersed systems.

Clear examples of poorly learnable phoneme inventories are the exaggerated and confusing and skewed inventories from §6, because they change into optimally dispersed systems. A direct measure of learnability could be the steepness of the black curve connecting the average produced frequency values over the generations: in a maximally learnable system, this is ∞, in a minimally learnable system it is much lower. In a phoneme inventory, as opposed to a morphosyntactic or semantic system, this steepness is severely constrained by the articulatory effect forcing categories towards the centre: categories cannot move from one extreme of the continuum to the other.

The steepness of a linear function $y = ax + b$ is a, the coefficient: $a = \Delta y/\Delta x$. If we are concerned with the learnability of the system of one generation, Δy equals 1; Δx is the shift on the auditory continuum, expressed here in the number of nodes. If we want to quantify the learnability of a system with a variable, say λ, and if we want optimally learnable systems to have a λ of 1 and maximally unlearnable systems a λ of 0, the simplest possible way to compute the λ of category c within a single generation is $\lambda_c = 1 - s_c$, where s_c is the relative shift s of category c on the auditory continuum, i.e. $|\Delta x|$ divided by the total number of nodes on the AudF layer.

The degree of learnability can be seen in light of the entire system: for instance, if /ʃ/ constitutes a single-category inventory, it has a lower λ than when it is as part of a two-category system /s ʃ/. (I adhere to the simplifying assumption that such systems are isolated and do not experience any influence from perceptually similar sounds from a different class, in the case of sibilants e.g. the (inter)dental fricative [θ] – see also B&H: 252, 264). Since we may have to average over multiple categories, a straightforward general formula for the learnability index λ of an entire system with n categories would be equation (2), averaging over all categories c:

(2) $$\lambda = \frac{1}{n}\sum_{c=1}^{n}(1 - s_c)$$

As noted above, phonological systems have a preference for the centre, so the value of s_c will always be lower than 1/2 in a one-category phonological inventory, usually lower than 1/3 in a two-category inventory, etc.; consequently, λ will be higher than 0.5 for any inventory. In fact, even in the exaggerated system of Figure 17 (p. 31), equation (2) yields a value of 0.7048 for generation 1, which is intuitively still quite high; and in Figure 16 (the “standard” distribution, p. 30), the learning curves in the first generation of learners are far from vertical, but still correspond to λ values close to 1 (0.9552 for category /A/, 0.9445 for /B/).

These values would be lower if we incorporate the degree by which a category can move through the auditory continuum at most, say $s_{c\max}$. In an n-category inventory, $s_{c\max} = 1/(n+1)$; dividing s_c by $s_{c\max}$, in order to get the relative relative (sic) shift, is mathematically equivalent to multiplying it by $n+1$, so if we do this, we obtain (3):

(3) $$\lambda = \frac{1}{n}\sum_{c=1}^{n}(1-(n+1)\cdot s_c)$$

Although this formula may still be too crude – e.g. $s_{c\max}$ depends on the parameter settings of the network; a category may move across (slightly) more than $1/(n+1)$ part of the auditory continuum, as long as its initial location is peripheral enough; the formula does not incorporate changes in standard deviations, so it would yield a low value of λ for an inventory with a bimodally distributed category –, it scales the λ values much better along the index range than equation (2) did. With (3), we obtain values of 0.1143 for the first generation of learners of the exaggerated distribution from Figure 17, and 0.9023 for the first generation of learners of the standard distribution from Figure 16.

Ceteris paribus, the average value of λ in any language with a non-standard initial distribution is expected to grow asymptotically towards 1 over a large number of generations. The probability of λ actually equalling 1 is minimal in any system, because the intergenerational transfer is noisy: for instance, there will always be transmission noise and random variation in production.

8 Conclusion, discussion and remaining issues

This thesis investigated whether artificial neural networks are a viable framework for modelling the evolution of auditory dispersion with the BiPhon model. The answer to this question seems to be 'yes', as all of B&H's findings with OT can be replicated in a neural network with the simplest layout possible. As in OT, a system with any type of initial distribution (standard, exaggerated, confusing, skewed, with or without a non-contiguous category) evolves into a stable inventory, striking an optimal balance between perceptual distinctiveness and articulatory ease. No teleological devices need to be included: the learning procedure does not have any tools to keep the auditory correlates of phonological categories distinct. Also, it does not have to compute any auditory distances between categories.

I have used both the outstar weight update rule (intended to optimize perception) and the inoutstar weight update rule (intended to optimize both perception and production, but not perfectly) in my simulations. Outstar learning predicts non-standard contrasts to be resolved much faster than inoutstar learning does; this latter rule predicts more gradient transitions into an optimally dispersed state than outstar learning does. Nevertheless, optimal auditory dispersion emerges with both types of learning. The advantage of inoutstar learning over outstar learning is that it can be used to replicate two unrelated phenomena at the phonology-phonetics interface, viz. auditory dispersion and the perceptual magnet effect.

To quantify the learnability of an inventory, i.e. the degree to which the auditory correlates of its categories are diachronically stable, I have proposed a learnability index λ, averaging over the relative shifts of the average auditory correlate of each category.

A logical next step in the strand of research presented here is the extension to the two-dimensional case, such as the vowel space, in which case Boersma (2007) would serve as inspiration. This will require significant adjustments to the network and the learning procedure, since only one AudF layer is to be assumed – after all, this is the spectral image of the incoming sound. This also has consequences for the computation of λ: in the OT-based formalization, the shift of a vowel category could easily be computed as the Euclidean distance between its coordinates in generation n and generation $n+1$. Within a single layer, this will be more complicated.

One of the major advantages of neural nets, their neurobiological validity, is also one of the most controversial issues surrounding them. While some researchers are concerned with accurate biological modelling, others focus on neural nets merely as a computational tool (e.g. Smolensky 1996). As a.o. Rumelhart & Zipser (1985: 98) point out, neural nets are usually crafted to do what they are intended to, without their creators giving heed to the question whether such a neural net could have evolved naturally; 25 years later, Gayler, Levy & Bod (2010) still see the need to urge researchers to create biologically plausible networks with functionally adequate layers.

One advantage of the present neural net over many others is the fact that it does not incorporate any hidden layers: the layers that it has, simplifying as they may be, all correspond to levels of representation that somehow have an analogue in humans. The ArtF layer embodies the entire articulatory system; the AudF layer relates to the auditory cortex, where the input from the basilar membrane is processed; the SF layer is analogous to a higher-level symbolic representation. Also, the present network aims to mirror the learning process as accurately as possible, by incorporating bioacoustical and environmental factors, such as activity spreading on the basilar membrane and the existence of transmission noise.

Still, some simplifications are made in this thesis, both in the architecture and workings of the network, and in the learning procedure. The sensorimotor connections are absent; phonological units are represented as a single node, while they are more likely to be distributed (i.e. activation patterns across multiple nodes); the learning rate does not decrease as the network matures; the activity summing in neurons may be more complex than it is in the present network; the language-acquiring infant does not face the task of perceptual learning with her lexicon fully in place; synapses are unidirectional, so bidirectional use of connections is not neurobiologically realistic; the neural net used here cannot yet represent more than one segment. Some of these simplifications are made deliberately, as I believe that they bear little influence on the results; others are more fundamental and deserve thorough further investigation. In general, more research is needed to successfully reproduce with neural nets all results achieved with OT, but neural nets prove to be a promising framework to cast phonetics and phonology in.

References

Ackley, David, Geoffrey Hinton & Terrence Sejnowski (1985). A learning algorithm for Boltzmann machines. *Cognitive Science* 9: 147–169.

Apoussidou, Diana (2007). *The learnability of metrical phonology*. PhD thesis, University of Amsterdam. Utrecht: LOT. http://www.fon.hum.uva.nl/theses/DianaApoussidouPhD2007.pdf

Azevedo, Frederico, Ludmila Carvalho, Lea Grinberg, José Marcelo Farfel, Renata Ferretti, Renata Leite, Wilson Jacob Filho, Roberto Lent & Suzana Herculano-Houzel (2009). Equal number of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *The Journal of Comparative Neurology* 513 (5): 532–541.

Blevins, Juliette (2004). *Evolutionary Phonology: the emergence of sound patterns*. Cambridge: Cambridge University Press.

Boersma, Paul (1998). *Functional Phonology: formalizing the interactions between articulatory and perceptual drives*. PhD thesis, University of Amsterdam. The Hague: Holland Academic Graphics. http://www.fon.hum.uva.nl/theses/PaulBoersmaPhD1998.pdf

Boersma, Paul (2001). Phonology-semantics interaction in OT, and its acquisition. In: Kirchner, Robert, Wolf Wikeley & Joe Pater (eds.), *Papers in experimental and theoretical linguistics, Volume 6*, 24–35. Edmonton: University of Alberta.

Boersma, Paul (2006). Prototypicality judgments as inverted perception. In: Fanselow, Gisbert, Caroline Féry, Ralf Vogel & Matthias Schlesewsky (eds.), *Gradience in grammar: generative perspectives*, 167–184. Oxford: Oxford University Press.

Boersma, Paul (2007). The emergence of auditory contrast. Presentation, GLOW 30. http://www.fon.hum.uva.nl/paul/presentations/VowelSpaceTromsoe2007.pdf

Boersma, Paul (2009a). *Emergent effects in simulations of language acquisition and evolution*. ACLC learnability lecture, University of Amsterdam, October 2, 2009.

Boersma, Paul (2009b). Cue constraints and their interactions in phonological perception and production. In: Boersma, Paul & Silke Hamann (eds.): *Phonology in perception*, 55–110. Berlin: Mouton de Gruyter.

Boersma, Paul (2011). A programme for bidirectional phonology and phonetics and their acquisition and evolution. In: Benz, Anton & Jason Mattausch (eds.), *Bidirectional Optimality Theory*, 33–72. Amsterdam: John Benjamins.

Boersma, Paul, Titia Benders & Klaas Seinhorst (fc.). *A neural network model for phonology and phonetics*.

Boersma, Paul, Paola Escudero & Rachel Hayes (2003). Learning abstract phonological from auditory phonetic categories: an integrated model for the acquisition of language-specific sound categories. *Proceedings of the 15th International Congress of Phonetic Sciences, Barcelona*, 1013–1016.

Boersma, Paul & Silke Hamann (2008). The evolution of auditory dispersion in bidirectional constraint grammars. *Phonology* 25: 217–270.

Boersma, Paul & David Weenink (2012). *Praat: doing phonetics by computer*. http://www.praat.org

Christiansen, Morten & Nick Chater (2008). Language as shaped by the brain. *Behavioral and Brain Sciences* 31: 489–558.

Clements, G. Nick (2003). Feature economy in sound systems. *Phonology* 20: 287–333.

Clements, G. Nick (2005). The role of features in speech sound inventories. In: Raimy, Eric & Charles Cairns (eds.), *Contemporary views on architecture and representations in phonological theory*, 19-68. Cambridge, MA: MIT Press.

de Boer, Bart (1999). *Self-organization in vowel systems*. PhD thesis, Free University Brussels. http://uvafon.hum.uva.nl/bart/papers/deBoerThesis.pdf de Boer, Bart (2000). Self-organization in vowel systems. *Journal of Phonetics* 28 (4): 441–465.

de Boer, Bart (2001). *The origins of vowel systems*. Oxford: Oxford University Press.

de Groot, Willem (1931). Phonologie und Phonetik als Funktionswissenschaften. *Travaux de Cercle Linguistique de Prague* 4: 146–147.

Guenther, Frank & Marin Gjaja (1996). The perceptual magnet effect as an emergent property of neural map formation. *Journal of the Acoustical Society of America* 100: 1111–1121.

Escudero, Paola & Paul Boersma (2004). Bridging the gap between L2 speech perception research and phonological theory. *Studies in Second Language Acquisition* 26: 551–585.

Flemming, Edward (1995/2002). *Auditory representations in phonology*. PhD dissertation, University of California, Los Angeles. Published by Routledge (London & New York).

Gayler, Ross, Simon Levy & Rens Bod (2010). Explanatory aspirations and the scandal of cognitive neuroscience. In: Samsonovich, Alexei, Kamilla Jóhannsdottir, Antonio Chella & Ben Goertzel (eds.), *Proceedings of the First Annual Meeting of the BICA Society*, 42–51. Washington D.C.: IOS Press.

Gazzaniga, Michael, Richard Ivry & George Mangun (2009). *Cognitive neuroscience: the biology of the mind*. New York: Norton. 3rd, revised edition.

Grossberg, Stephen (1969). Embedding fields: a theory of learning with physiological implications. *Journal of Mathematical Psychology* 6: 209–239.

Hall, Daniel Currie (2007). *The role and representation of contrast in phonological theory*. PhD thesis, University of Toronto. http://twpl.library.utoronto.ca/index.php/twpl/article/ view/6497

Hall, Daniel Currie (2011). Phonological contrast and its phonetic enhancement: dispersedness without dispersion. *Phonology* 28: 1–54.

Hamann, Silke (2009). The learner of a perception grammar as a source of sound change. In: Boersma, Paul & Silke Hamann (eds.), *Phonology in perception*, 111–150. Berlin/New York: Mouton de Gruyter.

Hooft, Pieter Corneliszoon (1617). *Warenar*. http://www.dbnl.org/tekst/hoof001ware01_01/

Hopfield, John (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America* 79 (8): 2554–2558.

Jakobson, Roman (1941). *Kindersprache, Aphasie, und allgemeine Lautgesetze*. Uppsala: Lundquist.

Jakobson, Roman & Morris Halle (1956). *Fundamentals of language*. The Hague: Mouton.

Johnson, Keith, Edward Flemming & Richard Wright (1993). The hyperspace effect: phonetic targets are hyperarticulated. *Language* 69 (3): 505–528.

Kirby, Simon (2001). Spontaneous evolution of linguistic structure: an iterated learning model of the emergence of regularity and irregularity. *IEEE Journal of Evolutionary Computation* 5: 102–110.

Kirby, Simon, Hannah Cornish & Kenny Smith (2008). Cumulative cultural evolution in the laboratory: an experimental approach to the origins of structure in language. *Proceedings of the National Academy of Sciences* 105 (31): 10681–10686.

Kirchner, Robert (1998). *An effort-based approach to consonant lenition*. PhD dissertation, University of California, Los Angeles. Rutgers Optimality Archive 276. Published in 2001, New York & London: Routledge.

Kuhl, Patricia (1991). Human adults and human infants show a “perceptual magnetic effect” for the prototypes of speech categories, monkeys do not. *Perception and Psychophysics* 50: 93–107.

Liljencrants, Johan & Björn Lindblom (1972). Numerical simulation of vowel quality systems: the role of perceptual contrast. *Linguistics* 48: 839–862.

Lindblom, Björn (1986). Phonetic universals in vowel systems. In: Ohala, John & Jeri Jaeger (eds.), *Experimental phonology*, 13–44. Orlando: Academic Press.

Lindblom, Björn & Ian Maddieson (1988). Phonetic universals in consonant systems. In: Hyman, Larry & Charles Li (eds.), *Language, speech and mind: studies in honour of Victoria A. Fromkin*, 62–78. London: Routledge.

Lopopolo, Alessandro (2011). *Population structure and acquisition dynamics influence language complexity*. MA thesis, rMA Linguistics, University of Amsterdam. http://www.fon.hum.uva.nl/theses/AlessandroLopopoloMA2011.pdf

Luce, Paul & David Pisoni (1998). Recognizing spoken words: the neighborhood activation model. *Ear and Hearing* 19: 1–36.

Lupyan, Gary & Rick Dale (2010). Language structure is partly determined by social structure. *PLoS ONE* 5 (1): e8559. doi:10.1371/journal.pone.0008559

Martinet, André (1960). *Eléments de linguistique générale*. Paris: Armand Colin.

McCarthy, John & Alan Prince (1995). Faithfulness and reduplicative identity. In: Beckman, Jill, Laura Walsh Dickey & Suzanne Urbanczyk (eds.), *Papers in Optimality Theory*, 249–384. University of Massachusetts Occasional Papers 18. Amherst, MA: Graduate Linguistic Student Association.

McCulloch, Warren & Walter Pitts (1943). A logical calculus of the ideas immanent in neural nets. *Bulletin of Mathematical Biophysics* 5: 115–137.

Mielke, Jeff (2004). *The emergence of distinctive features*. PhD dissertation, Ohio State University. Published in 2008, Oxford: Oxford University Press.

Mielke, Jeff (2008). *P-base*. Database of sound patterns. http://aix1.uottawa.ca/~jmielke/pbase/index.html, http://137.122.133.199/cgi-bin/pbase3/search.cgi

Mitchell, Tom (1997). *Machine learning*. New York etc.: McGraw-Hill.

Moore, Brian & Brian Glasberg (1983). Suggested formulae for calculating auditory-filter bandwidths and excitation patterns. *Journal of the Acoustical Society of America* 74 (3): 750–753.

Ohala, John (1980). Moderator's summary of 'Symposium no.1: phonetic universals in phonological systems and their explanation'. In: *Proceedings of the 9th International Congress of Phonetic Sciences, Institute of Phonetics, University of Copenhagen*, 181–194. http://linguistics.berkeley.edu/~ohala/papers/sympos_on_universals_icphs_copenhagen.pdf

Ohala, John (1981). The listener as a source of sound change. In: Masek, Carrie, Roberta Hendrick & Mary Frances Miller (eds.), *Papers from the parasession on language and behavior*, 178–203. Chicago: Chicago Linguistic Society.

Padgett, Jaye (2001). Contrast dispersion and Russian palatalization. In: Hume, Elizabeth & Keith Johnson (eds.), *The role of speech perception in phonology*, 187–218. San Diego: Academic Press.

Padgett, Jaye (2003). Contrast and post-velar fronting in Russian. *Natural Language & Linguistic Theory* 21: 39–87.

Padgett, Jaye & Marzena Żygis (2003). The evolution of sibilants in Polish and Russian. In: Hall, Tracy Alan & Silke Hamann (eds.), *ZAS Papers in Linguistics* 32, 155–174. http://people.ucsc.edu/~padgett/locker/sibilantpaper03.pdf Reprinted in 2007, *Journal of Slavic Linguistics* 15 (2): 291–324.

Passy, Paul (1890). *Etude sur les changements phonétiques et leur caractères généraux*. Paris: Firmin-Didot.

Prince, Alan & Paul Smolensky (1993/2004). *Optimality Theory: constraint interaction in Generative Grammar*. ROA-1 Blackwell.

Roth, Gerhard & Ursula Dicke (2005). Evolution of the brain and intelligence. *Trends in Cognitive Science* 9 (5): 250–257.

Rumelhart, David and David Zipser (1985). Feature discovery by competitive learning. *Cognitive Science* 9: 75–112.

Sanders, Nathan (2003). *Opacity and sound change in the Polish lexicon*. PhD thesis, University of California, Santa Cruz. http://sanders.phonologist.org/diss.html

Schwartz, Jean-Luc, Louis-Jean Boë, Nathalie Vallée & Christian Abry (1997). Major trends in vowel system inventories. *Journal of Phonetics* 25: 233–253.

Seinhorst, Klaas & Etske Ooijevaar (2011). *The auditory dispersion of Dutch sibilants*. Presentation, Workshop "Windows on language change and learnability", University of Amsterdam.

Smolensky, Paul (1996). Overview: computational, dynamical, and statistical perspectives on the processing and learning problems in neural network theory. In: Smolensky, Paul, Michael Mozer & David Rumelhart (eds.), *Mathematical perspectives on neural networks*, 1–16. New Jersey: Lawrence Erlbaum.

Smolensky, Paul & Géraldine Legendre (2006). *The harmonic mind: from neural computation to optimality-theoretic grammar*. Cambridge, MA: MIT Press.

Stevens, Kenneth (1972). The quantal nature of speech: evidence from articulatory-acoustic data. In: Denes, Peter & Edward David Jr (eds.), *Human communication: a unified view*, 51–66. New York [etc.]: McGraw-Hill.

Stevens, Kenneth (1989). On the quantal nature of speech. *Journal of Phonetics* 17: 3–46.

Stevens, Kenneth & Samuel Keyser (2010). Quantal theory, enhancement and overlap. *Journal of Phonetics* 38: 10–19.

Stieber, Ździsław (1952). *Rozwój języka polskiego*. Warsaw: Państwowe Wydawnictwo Naukowe.

Stieber, Ździsław (1973). *A historical phonology of the Polish language*. Heidelberg: Winter.

ten Bosch, Louis (1991). *On the structure of vowel systems: aspects of an extended vowel model using effort and contrast*. PhD thesis, University of Amsterdam. Helmond: Wibro Dissertatiedrukkerij.

Theisen, Carrie Ann, Jon Oberlander & Simon Kirby (2010). Systematicity and arbitrariness in novel communication systems. *Interaction Studies* 11 (1): 14–32.

van der Sijs, Nicoline (2002). *Chronologisch woordenboek. De ouderdom en herkomst van onze woorden en betekenissen*. Amsterdam/Antwerp: Veen. 2nd edition.

van Leussen, Jan-Willem (2008). *Emergent optimal vowel systems*. MA thesis, General Linguistics, University of Amsterdam. http://www.fon.hum.uva.nl/theses/JanWillemVanLeussenMA2008.pdf

van Leussen, Jan-Willem (2009). *Investigating consonant inventories with acoustic and articulatory models*. MSc thesis, Artificial Intelligence, University of Amsterdam. http://www.fon.hum.uva.nl/theses/JanWillemVanLeussenMA2009.pdf

van Leussen, Jan-Willem & Maaike Vondenhoff (2008). *A self-organizational approach to the gradual learning algorithm in modelling the evolution of auditory contrast*. Term paper, University of Amsterdam. http://www.fon.hum.uva.nl/janwillem/papers/EoS_Maaike_JW.pdf

Verhoef, Tessa & Bart de Boer (2011a). Cultural emergence of feature economy in an artificial whistled language. In: Zee, Eric & Wai-Sum Lee (eds.), *Proceedings of the ICPhS XVII*, 2066–2069. Hong Kong: City University of Hong Kong. http://www.icphs2011.hk/resources/OnlineProceedings/RegularSession/Verhoef/Verhoef.pdf

Verhoef, Tessa & Bart de Boer (2011b). Language acquisition age effects and their role in the preservation and change of communication systems. *Linguistics in Amsterdam* 4 (1). http://dare.uva.nl/document/344592 von der Gabelentz, Georg (1901). *Die Sprachwissenschaft: ihre Aufgaben, Methoden und bisherigen Ergebnisse*. Leipzig: Tauchnitz.

Wanrooij, Karin (2009). *Does distributional input improve the categorization of speech sounds? Neurobiological aspects and computer simulations*. MA thesis, Linguistics, University of Amsterdam. http://www.fon.hum.uva.nl/theses/KarinWanrooijMA2009.pdf

Wedel, Andrew (2006). Exemplar models, evolution and language change. *The Linguistic Review* 23: 247–274. http://dingo.sbs.arizona.edu/~wedel/publications/PDF/WedelExEv_LangChange.pdf

Williams, Robert & Karl Herrup (1988). The control of neuron number. *Annual Review of Neuroscience* 11: 423–453.

Zipf, George (1935). *The psychobiology of language: an introduction to dynamic philology*. New York: Houghton-Mifflin.

Zipf, George (1949). *Human behavior and the principle of least effort: an introduction to human ecology*. Cambridge, MA: Addison-Wesley.

Żygis, Marzena & Jaye Padgett (2007). A perceptual study of Polish fricatives, and its implications for historical sound change. *Journal of Phonetics* 38: 207–226.

Appendix A: The evolution of auditory dispersion in an inventory with contiguous categories

```
# Praat-script Evolution of auditory dispersion (unimodal)
# klaas seinhorst, 10 August 2012. Parts of this script were adapted from a script by Paul Boersma.

form Auditory dispersion
    real Mean_of_category_1 0.35
    real Mean_of_category_2 0.65
    real Mean_of_category_3 0
    real Mean_of_category_4 0
    natural Number_of_generations 10
    natural Number_of_steps 100000
    real Transmission_noise 0.05
    natural Number_of_Auditory_Forms 100
    boolean Articulatory_Form 1
    real Articulatory_inhibition 1.5
    real Weight_of_central_connection -0.25
    real Instar 0
    real Outstar 1
    real Temperature 0.1
    real Learning_rate 0.001
    choice Activation_clipping 3
        option linear
        option sigmoid
        option top-sigmoid
endform

if mean_of_category_4 <> 0
    sf.numberOfNodes = 4
elsif mean_of_category_3 <> 0
    sf.numberOfNodes = 3
elsif mean_of_category_2 <> 0
    sf.numberOfNodes = 2
else
    sf.numberOfNodes = 1
endif
for i to sf.numberOfNodes
    meanOfCategory'i' = mean_of_category_'i'
endfor
audf.numberOfNodes = number_of_Auditory_Forms
artConnStrength = articulatory_inhibition
weightOfCentrArtConn = weight_of_central_connection
transmissionNoise = transmission_noise * audf.numberOfNodes
spreadingOfAmbient = audf.numberOfNodes / 10

#
# Levels, nodes and connections.
#
if articulatory_Form = 1
    artf.numberOfNodes = 1
    artf.offsetNode = 0
    artf.offsetConnection = 0
    audf.offsetNode = artf.offsetNode + artf.numberOfNodes
    audf.offsetConnection = artf.offsetConnection + artf.numberOfNodes
    artf.y = 1.0
    audf.y = 5.0
    sf.y = 9.0
else
    artf.numberOfNodes = 0
    audf.offsetNode = 0
    audf.offsetConnection = 0
    audf.y = 2.0
    sf.y = 8.0
```

```
endif
sf.offsetNode = audf.offsetNode + audf.numberOfNodes
net.numberOfNodes = sf.offsetNode + sf.numberOfNodes

audf.numberOfConnections = 0
cue.numberOfConnections = sf.numberOfNodes * audf.numberOfNodes
cue.offsetConnection = audf.offsetConnection + audf.numberOfConnections
sf.numberOfConnections = 0
sf.offsetConnection = cue.offsetConnection + cue.numberOfConnections

#
# Default properties of the network.
#
net.amin = 0   ; minimum activation level for unclamped nodes
net.amax = 1   ; maximum activation level for unclamped nodes
net.activityLeak = 0
net.spreadingRate = 1
net.wmin = -10   ; minimum weight of a connection (negative to allow inhibition)
net.wmax = 10   ; maximum weight of a connection (positive to allow excitation)
net.weightLeak = 0
net.xmin = 0   ; for drawing
net.xmax = 10   ; for drawing
net.ymin = 0   ; for drawing
net.ymax = 10   ; for drawing
net.initialWmin = 0
net.initialWmax = 0.25

#
# Network creation.
#
procedure createNetwork
    network = Create empty Network... dispersion net.spreadingRate 'activation_clipping$'
    ... net.amin net.amax net.activityLeak learning_rate net.wmin net.wmax net.weightLeak
    ... net.xmin net.xmax net.ymin net.ymax
    Set instar... instar
    Set outstar... outstar
    if articulatory_Form = 1
        for .i to artf.numberOfNodes
            Add node... net.xmin+(net.xmax-net.xmin)/artf.numberOfNodes*(.i-0.5) artf.y 1 no
        endfor
        for .i to audf.numberOfNodes
            a = 4 * artConnStrength / (audf.numberOfNodes ^ 2 - 1)
            b = (1 + audf.numberOfNodes) / 2
            Add connection... artf.offsetNode+1 audf.offsetNode+.i weightOfCentrArtConn-a*(.i-b)^2 0
        endfor
    endif
    for .i to audf.numberOfNodes
        Add node... net.xmin+(net.xmax-net.xmin)/audf.numberOfNodes*(.i-0.5) audf.y 0
randomUniform(net.amin,net.amax) no
    endfor
    for .i to sf.numberOfNodes
        Add node... net.xmin+(net.xmax-net.xmin)/sf.numberOfNodes*(.i-0.5) sf.y
randomUniform(net.amin,net.amax) no
    endfor
    for .i to sf.numberOfNodes
        for .j to audf.numberOfNodes
            Add connection...  audf.offsetNode+.j sf.offsetNode+.i
randomUniform(net.initialWmin,net.initialWmax) 1
        endfor
    endfor
endproc

#
# The initial language.
#
```

```
stdevOfCategory = audf.numberOfNodes / 16
Create Table with column names... OutputProbabilitiesGen0 audf.numberOfNodes node
Formula... node row
for activeCategory to sf.numberOfNodes
    Append column... a'activeCategory'
    meanNodeOfCategory = meanOfCategory'activeCategory' * (audf.numberOfNodes + 1)
    Formula... a'activeCategory' exp (-0.5 * (self ["node"] - meanNodeOfCategory) ^ 2 / stdevOfCategory ^ 2)
    cumul'activeCategory' = 0
    for i to audf.numberOfNodes
        act = Get value... i a'activeCategory'
        cumul'activeCategory' += act
    endfor
endfor


for generation to number_of_generations + 1
    call createNetwork
    Create Table with column names... OutputActivitiesGen'generation' audf.numberOfNodes node
    Formula... node row
    for i to sf.numberOfNodes
        Append column... a'i'
    endfor
    Copy... PerceptionGen'generation'
    Create Table with column names... InputBeforeNoiseGen'generation' number_of_steps category node
    Copy... InputAfterNoiseGen'generation'
    select network
    #
    # Learn perception.
    #
    parent = generation - 1
    for step to number_of_steps
        Zero activities... 0 0
        activeCategory = randomInteger (1, sf.numberOfNodes)   ; /A/ = 1, /B/ = 2 etc.
        sf = sf.offsetNode + activeCategory
        Set activity... sf 1
        select Table OutputProbabilitiesGen'parent'
        p = randomUniform (0, cumul'activeCategory')
        producedNode = 0
        sum = 0
        repeat
            producedNode += 1
            act = Get value... producedNode a'activeCategory'
            sum += act
        until sum >= p
        select Table InputBeforeNoiseGen'generation'
        Set numeric value... step category activeCategory
        Set numeric value... step node producedNode
        select network
        perceivedNode = randomGauss (producedNode, transmissionNoise)
        if perceivedNode < 1
            perceivedNode = 1
        elsif perceivedNode > audf.numberOfNodes
            perceivedNode = audf.numberOfNodes
        endif
        select Table InputAfterNoiseGen'generation'
        Set numeric value... step category activeCategory
        Set numeric value... step node perceivedNode
        select network
        for i to audf.numberOfNodes
            activity = exp (-0.5 * (i - perceivedNode) ^ 2 / (spreadingOfAmbient ^ 2))
            Set activity... audf.offsetNode+i activity
        endfor
        for i from audf.offsetNode+1 to net.numberOfNodes
            Set clamping... i yes
        endfor
```

```
        Update weights
    endfor
    #
    # Compute the in- and output statistics.
    #
    Create Table with column names... InputStatsBeforeNoiseGen'generation' sf.numberOfNodes category
mean stdev
    Formula... category row
    Copy... InputStatsAfterNoiseGen'generation'
    for activeCategory to sf.numberOfNodes
        select Table InputBeforeNoiseGen'generation'
        Extract rows where column (number)... category "equal to" activeCategory
        select Table InputBeforeNoiseGen'generation'__category__'activeCategory'
        mean = Get mean... node
        stdev = Get standard deviation... node
        Sort rows... node
        select Table InputStatsBeforeNoiseGen'generation'
        Set numeric value... activeCategory mean mean
        Set numeric value... activeCategory stdev stdev
        select Table InputAfterNoiseGen'generation'
        Extract rows where column (number)... category "equal to" activeCategory
        select Table InputAfterNoiseGen'generation'__category__'activeCategory'
        mean = Get mean... node
        stdev = Get standard deviation... node
        Sort rows... node
        select Table InputStatsAfterNoiseGen'generation'
        Set numeric value... activeCategory mean mean
        Set numeric value... activeCategory stdev stdev
    endfor
    #
    # Determine the cue connection weights.
    #
    select network
    for audf from audf.offsetNode+1 to sf.offsetNode
        Zero activities... 0 0
        for i from audf.offsetNode+1 to sf.offsetNode
            Set clamping... i yes
        endfor
        for i to sf.numberOfNodes
            Set clamping... sf.offsetNode+i no
        endfor
        Set activity... audf 1
        Spread activities... 1
        for i to sf.numberOfNodes
            perceive'i' = Get activity... sf.offsetNode+i
            select Table PerceptionGen'generation'
            if articulatory_Form = 1
                Set numeric value... audf-audf.offsetNode a'i' perceive'i'
            else
                Set numeric value... audf a'i' perceive'i'
            endif
            select network
        endfor
    endfor
    #
    # Speak.
    #
    for sf from sf.offsetNode+1 to net.numberOfNodes
        Zero activities... 0 0
        activeCategory = sf - sf.offsetNode
        for i from sf.offsetNode+1 to net.numberOfNodes
            Set clamping... i yes
        endfor
        for i from audf.offsetNode+1 to sf.offsetNode
            Set clamping... i no
```

```
        endfor
        Set activity... sf 1
        if articulatory_Form = 1
            Set clamping... 1 yes
            Set activity... 1 1
        endif
        Spread activities... 1
        for i to audf.numberOfNodes
            activity = Get activity... audf.offsetNode+i
            select Table OutputActivitiesGen'generation'
            Set numeric value... i a'activeCategory' exp (activity / temperature)
            select network
        endfor
    endfor
    select Table OutputActivitiesGen'generation'
    Copy... OutputProbabilitiesGen'generation'
    for activeCategory to sf.numberOfNodes
        max'activeCategory' = Get maximum... a'activeCategory'
        Formula... a'activeCategory' (self) / max'activeCategory'
        cumul'activeCategory' = 0
        for i to audf.numberOfNodes
            act = Get value... i a'activeCategory'
            cumul'activeCategory' += act
        endfor
    endfor
    #
    # Go forth and multiply.
    #
    select network
    Remove
endfor

#
# Draw.
#
Erase all
Select outer viewport... 0 6 0.25 3.5
Times
12
Colour... Black
Line width... 1
Solid line
Axes... 1 audf.numberOfNodes number_of_generations+1 1
Draw inner box
Text left... 1 generation
Text bottom... 1 AudF node
Marks bottom... 2 yes no no
Marks bottom... 11 no yes yes
Marks left... number_of_generations+1 no yes yes
for generation to number_of_generations
    Text... audf.numberOfNodes/-30 centre generation+0.5 half 'generation'
endfor
Create Table with column names... evolution number_of_generations+1 generation
Formula... generation row
for category to sf.numberOfNodes
    Append column... mean'category'
    Append column... stdev'category'
endfor

for generation to number_of_generations+1
    for category to sf.numberOfNodes
        select Table InputStatsBeforeNoiseGen'generation'
        mean = Get value... category mean
        stdev = Get value... category stdev
        select Table evolution
```

```
        Set numeric value... generation mean'category' mean
        Set numeric value... generation stdev'category' stdev
    endfor
endfor

for generation from 1 to number_of_generations
    for category to sf.numberOfNodes
        child = generation + 1
        meanGeneration = Get value... generation mean'category'
        meanChild = Get value... child mean'category'
        stdevGeneration = Get value... generation stdev'category'
        stdevChild = Get value... child stdev'category'
        Line width... 3
        Draw line... meanGeneration generation meanChild child
        Colour... Grey
        Line width... 2
        Draw line... meanGeneration-stdevGeneration generation meanChild-stdevChild child
        Draw line... meanGeneration+stdevGeneration generation meanChild+stdevChild child
        Colour... Black
    endfor
endfor

Line width... 1

select Table evolution
plus Table OutputProbabilitiesGen0
Remove
for generation to number_of_generations+1
    select Table InputAfterNoiseGen'generation'
    plus Table InputBeforeNoiseGen'generation'
    plus Table InputStatsAfterNoiseGen'generation'
    plus Table InputStatsBeforeNoiseGen'generation'
    plus Table OutputActivitiesGen'generation'
    plus Table OutputProbabilitiesGen'generation'
    plus Table PerceptionGen'generation'
    Remove
endfor
for generation to number_of_generations+1
    for category to sf.numberOfNodes
        select Table InputAfterNoiseGen'generation'__category__'category'
        plus Table InputBeforeNoiseGen'generation'__category__'category'
        Remove
    endfor
endfor
```

Appendix B: The evolution of auditory dispersion in an inventory with one non-contiguous category

```
# Praat-script Evolution of auditory dispersion (bimodal)
# klaas seinhorst, 10 August 2012. Parts of this script were adapted from a script by Paul Boersma.

form Auditory dispersion (bimodal)
    real Peak_of_unimodal_category 0.5
    real Peak_1_of_bimodal_category 0.3125
    real Peak_2_of_bimodal_category 0.6875
    real Ratio 3
    # (how much larger is the probability of the right peak in the initial distribution?)
    natural Number_of_generations 10
    natural Number_of_steps 100000
    real Transmission_noise 0.05
    natural Number_of_Auditory_Forms 100
    boolean Articulatory_Form 1
    real Articulatory_inhibition 1.5
    real Weight_of_central_connection -0.25
    real Instar 0
    real Outstar 1
    real Temperature 0.1
    real Learning_rate 0.001
    choice Activation_clipping 3
        option linear
        option sigmoid
        option top-sigmoid
endform

sf.numberOfNodes = 2
audf.numberOfNodes = number_of_Auditory_Forms
transmissionNoise = transmission_noise * audf.numberOfNodes
spreadingOfAmbient = audf.numberOfNodes / 10
peakOfUnimodal = peak_of_unimodal_category
for i to 2
    peak'i'OfBimodal = peak_'i'_of_bimodal_category
endfor
artConnStrength = articulatory_inhibition
weightOfCentrArtConn = weight_of_central_connection

#
# Levels, nodes and connections.
#
if articulatory_Form = 1
    artf.numberOfNodes = 1
    artf.offsetNode = 0
    artf.offsetConnection = 0
    audf.offsetNode = artf.offsetNode + artf.numberOfNodes
    audf.offsetConnection = artf.offsetConnection + artf.numberOfNodes
    artf.y = 1.0
    audf.y = 5.0
    sf.y = 9.0
else
    artf.numberOfNodes = 0
    audf.offsetNode = 0
    audf.offsetConnection = 0
    audf.y = 2.0
    sf.y = 8.0
endif
sf.offsetNode = audf.offsetNode + audf.numberOfNodes
net.numberOfNodes = sf.offsetNode + sf.numberOfNodes

audf.numberOfConnections = 0
```

```
cue.numberOfConnections = sf.numberOfNodes * audf.numberOfNodes
cue.offsetConnection = audf.offsetConnection + audf.numberOfConnections
sf.numberOfConnections = 0
sf.offsetConnection = cue.offsetConnection + cue.numberOfConnections

#
# Default properties of the network.
#
net.amin = 0   ; minimum activation level for unclamped nodes
net.amax = 1   ; maximum activation level for unclamped nodes
net.activityLeak = 0
net.spreadingRate = 1
net.wmin = -10   ; minimum weight of a connection (negative to allow inhibition)
net.wmax = 10   ; maximum weight of a connection (positive to allow excitation)
net.weightLeak = 0
net.xmin = 0   ; for drawing
net.xmax = 10   ; for drawing
net.ymin = 0   ; for drawing
net.ymax = 10   ; for drawing
net.initialWmin = 0
net.initialWmax = 0.25

#
# Network creation.
#
procedure createNetwork
    network = Create empty Network... dispersion net.spreadingRate 'activation_clipping$'
    ... net.amin net.amax net.activityLeak learning_rate net.wmin net.wmax net.weightLeak
    ... net.xmin net.xmax net.ymin net.ymax
    Set instar... instar
    Set outstar... outstar
    if articulatory_Form = 1
        for .i to artf.numberOfNodes
            Add node... net.xmin+(net.xmax-net.xmin)/artf.numberOfNodes*(.i-0.5) artf.y 1 no
        endfor
        for .i to audf.numberOfNodes
            a = 4 * artConnStrength / (audf.numberOfNodes ^ 2 - 1)
            b = (1 + audf.numberOfNodes) / 2
            Add connection... artf.offsetNode+1 audf.offsetNode+.i weightOfCentrArtConn-a*(.i-b)^2 0
        endfor
    endif
    for .i to audf.numberOfNodes
        Add node... net.xmin+(net.xmax-net.xmin)/audf.numberOfNodes*(.i-0.5) audf.y 0
randomUniform(net.amin,net.amax) no
    endfor
    for .i to sf.numberOfNodes
        Add node... net.xmin+(net.xmax-net.xmin)/sf.numberOfNodes*(.i-0.5) sf.y
randomUniform(net.amin,net.amax) no
    endfor
    for .i to sf.numberOfNodes
        for .j to audf.numberOfNodes
            Add connection...  audf.offsetNode+.j sf.offsetNode+.i
randomUniform(net.initialWmin,net.initialWmax) 1
        endfor
    endfor
endproc

procedure saveTable fileName$
    select Table 'fileName$'
    Save as tab-separated file... 'fileName$'.Table
endproc

#
# The initial language.
#
```

```
meanNode1OfBimodal = peak1OfBimodal * (audf.numberOfNodes + 1)
meanNode2OfBimodal = peak2OfBimodal * (audf.numberOfNodes + 1)
meanNodeOfUnimodal = peakOfUnimodal * (audf.numberOfNodes + 1)
stdevOfCategory = audf.numberOfNodes / 16
Create Table with column names... OutputProbabilitiesGen0 audf.numberOfNodes node a1 a2
Formula... node row
Formula... a1 exp (-0.5 * (self ["node"] - meanNodeOfUnimodal) ^ 2 / stdevOfCategory ^ 2)
Formula... a2 (1 / (ratio + 1)) * exp (-0.5 * (self ["node"] - meanNode1OfBimodal) ^ 2 / stdevOfCategory ^ 2) +
... (ratio / (ratio + 1)) * exp (-0.5 * (self ["node"] - meanNode2OfBimodal) ^ 2 / stdevOfCategory ^ 2)
for activeCategory to sf.numberOfNodes
    cumul'activeCategory' = 0
    for i to audf.numberOfNodes
        act = Get value... i a'activeCategory'
        cumul'activeCategory' += act
    endfor
endfor

for generation to number_of_generations + 1
    call createNetwork
    Create Table with column names... OutputActivitiesGen'generation' audf.numberOfNodes node
    Formula... node row
    for i to sf.numberOfNodes
        Append column... a'i'
    endfor
    Copy... PerceptionGen'generation'
    Create Table with column names... InputBeforeNoiseGen'generation' number_of_steps category node
    Copy... InputAfterNoiseGen'generation'
    select network
    #
    # Learn perception.
    #
    parent = generation - 1
    for step to number_of_steps
        Zero activities... 0 0
        activeCategory = randomInteger (1, sf.numberOfNodes)   ; /A/ = 1, /B/ = 2 etc.
        sf = sf.offsetNode + activeCategory
        Set activity... sf 1
        select Table OutputProbabilitiesGen'parent'
        p = randomUniform (0, cumul'activeCategory')
        producedNode = 0
        sum = 0
        repeat
            producedNode += 1
            act = Get value... producedNode a'activeCategory'
            sum += act
        until sum >= p
        select Table InputBeforeNoiseGen'generation'
        Set numeric value... step category activeCategory
        Set numeric value... step node producedNode
        select network
        perceivedNode = randomGauss (producedNode, transmissionNoise)
        if perceivedNode < 1
            perceivedNode = 1
        elsif perceivedNode > audf.numberOfNodes
            perceivedNode = audf.numberOfNodes
        endif
        select Table InputAfterNoiseGen'generation'
        Set numeric value... step category activeCategory
        Set numeric value... step node perceivedNode
        select network
        for i to audf.numberOfNodes
            activity = exp (-0.5 * (i - perceivedNode) ^ 2 / (spreadingOfAmbient ^ 2))
            Set activity... audf.offsetNode+i activity
        endfor
        for i from audf.offsetNode+1 to net.numberOfNodes
```

```
            Set clamping... i yes
        endfor
        Update weights
    endfor
    #
    # Compute the in- and output statistics.
    #
    Create Table with column names... InputStatsBeforeNoiseGen'generation' sf.numberOfNodes category
mean stdev
    Formula... category row
    Copy... InputStatsAfterNoiseGen'generation'
    for activeCategory to sf.numberOfNodes
        select Table InputBeforeNoiseGen'generation'
        Extract rows where column (number)... category "equal to" activeCategory
        select Table InputBeforeNoiseGen'generation'__category__'activeCategory'
        mean = Get mean... node
        stdev = Get standard deviation... node
        Sort rows... node
        select Table InputStatsBeforeNoiseGen'generation'
        Set numeric value... activeCategory mean mean
        Set numeric value... activeCategory stdev stdev
        select Table InputAfterNoiseGen'generation'
        Extract rows where column (number)... category "equal to" activeCategory
        select Table InputAfterNoiseGen'generation'__category__'activeCategory'
        mean = Get mean... node
        stdev = Get standard deviation... node
        Sort rows... node
        select Table InputStatsAfterNoiseGen'generation'
        Set numeric value... activeCategory mean mean
        Set numeric value... activeCategory stdev stdev
    endfor
    select Table InputBeforeNoiseGen'generation'
    plus Table InputAfterNoiseGen'generation'
    Remove
    #
    # Determine the cue connection weights.
    #
    select network
    for audf from audf.offsetNode+1 to sf.offsetNode
        Zero activities... 0 0
        for i from audf.offsetNode+1 to sf.offsetNode
            Set clamping... i yes
        endfor
        for i to sf.numberOfNodes
            Set clamping... sf.offsetNode+i no
        endfor
        Set activity... audf 1
        Spread activities... 1
        for i to sf.numberOfNodes
            perceive'i' = Get activity... sf.offsetNode+i
            select Table PerceptionGen'generation'
            if articulatory_Form = 1
                Set numeric value... audf-audf.offsetNode a'i' perceive'i'
            else
                Set numeric value... audf a'i' perceive'i'
            endif
            select network
        endfor
    endfor
    #
    # Speak.
    #
    for sf from sf.offsetNode+1 to net.numberOfNodes
        Zero activities... 0 0
        activeCategory = sf - sf.offsetNode
```

```
        for i from sf.offsetNode+1 to net.numberOfNodes
            Set clamping... i yes
        endfor
        for i from audf.offsetNode+1 to sf.offsetNode
            Set clamping... i no
        endfor
        Set activity... sf 1
        if articulatory_Form = 1
            Set clamping... 1 yes
            Set activity... 1 1
        endif
        Spread activities... 1
        for i to audf.numberOfNodes
            activity = Get activity... audf.offsetNode+i
            select Table OutputActivitiesGen'generation'
            Set numeric value... i a'activeCategory' exp (activity / temperature)
            select network
        endfor
    endfor
    select Table OutputActivitiesGen'generation'
    Copy... OutputProbabilitiesGen'generation'
    for activeCategory to sf.numberOfNodes
        max'activeCategory' = Get maximum... a'activeCategory'
        Formula... a'activeCategory' (self) / max'activeCategory'
        cumul'activeCategory' = 0
        for i to audf.numberOfNodes
            act = Get value... i a'activeCategory'
            cumul'activeCategory' += act
        endfor
    endfor
    #
    # Go forth and multiply.
    #
    select network
    Remove
endfor

#
# Draw.
#
Erase all
Select outer viewport... 0 6 0.25 3.5
Times
12
Colour... Black
Line width... 1
Solid line
Axes... 1 audf.numberOfNodes number_of_generations+1 1
Draw inner box
Text left... 1 generation
Text bottom... 1 AudF node
Marks bottom... 2 yes no no
Marks bottom... 9 no yes yes
Marks left... number_of_generations+1 no yes yes
for generation to number_of_generations
    Text... audf.numberOfNodes/-30 centre generation+0.5 half 'generation'
endfor
Create Table with column names... evolution number_of_generations+1 generation
Formula... generation row
for category to 2
    Append column... mean'category'
    Append column... stdev'category'
endfor

for generation to number_of_generations+1
```

```
    for category to 2
        select Table InputStatsBeforeNoiseGen'generation'
        mean = Get value... category mean
        stdev = Get value... category stdev
        select Table evolution
        Set numeric value... generation mean'category' mean
        Set numeric value... generation stdev'category' stdev
    endfor
endfor

for generation from 1 to number_of_generations
    for category to 2
        if category = 1
            Solid line
        else
            Dashed line
        endif
        child = generation + 1
        meanGeneration = Get value... generation mean'category'
        meanChild = Get value... child mean'category'
        stdevGeneration = Get value... generation stdev'category'
        stdevChild = Get value... child stdev'category'
        Line width... 3
        Draw line... meanGeneration generation meanChild child
        Line width... 2
        Draw line... meanGeneration-stdevGeneration generation meanChild-stdevChild child
        Draw line... meanGeneration+stdevGeneration generation meanChild+stdevChild child
    endfor
endfor

Solid line
Line width... 1

select Table evolution
plus Table OutputProbabilitiesGen0
Remove
for generation to number_of_generations+1
    select Table InputStatsAfterNoiseGen'generation'
    plus Table InputStatsBeforeNoiseGen'generation'
    plus Table OutputActivitiesGen'generation'
    plus Table OutputProbabilitiesGen'generation'
    plus Table PerceptionGen'generation'
    Remove
endfor
for generation to number_of_generations+1
    for category to sf.numberOfNodes
        select Table InputAfterNoiseGen'generation'__category__'category'
        plus Table InputBeforeNoiseGen'generation'__category__'category'
        Remove
    endfor
endfor
```