Speech Technology Project: Unsupervised Distributional Learning of Vowel Contrasts in L2 Acquisition

Stijn van Balen (9946667), Jan-Willem van Leussen (0240184), and Andrea Schuch (0639958)

October 3, 2007

Abstract

This report describes a project on unsupervised learning of L2 vowel categories by three human learners and two learning algorithms. To test the learning progress we used a game-like testing tool to provide the learning samples and assess the categorization of vowels before and after the learning process. The two algorithms used were Time Driven Kernel Estimation (TDKE), developed by one of the authors, and Optimality Theory using the Gradual Learning Algorithm. Testing results suggest that there is some post-training improvement for the human learners and the TDKE, but not for the Gradual Learning Algorithm.

Contents

1 Introduction **2**

2 Experiments **3**

3 Simulation **4**

3.1 Kohonen Self-Organizing Map 5

3.1.1 Description of the Algorithms 7

3.1.2 Result of the Simulation 8

3.2 Optimality Theory 10

3.2.1 Modeling a Beginning L2 Learner 10

3.2.2 Modeling the Testing and Training 11

3.3 A Time Driving Kernel Estimation Algorithm 11

3.3.1 The Mechanics 12

3.3.2 The Algorithm 15

3.3.3 The Characteristics 16

3.3.4 The Results 16

4 Results and Analysis **17**

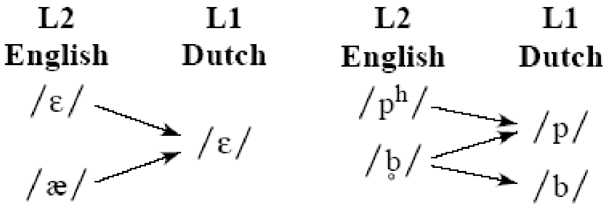


Figure 1: Examples of single-category assimilation (left) and two-category assimilation (left) by [5].

5 Conclusions and Future Work **19**

1 Introduction

We model the perceptual acquisition of speech sound categories in second language learning of adults. Second-language speech perception is different from first language acquisition (of infants): While children are "born with general auditory mechanisms to process all speech sounds of human languages" [8], we assume that our learners have already lost this "universal" sensitivity, as they have specialized the categories of their native language. As a consequence, "L2 learners have difficulties when perceiving phonological contrasts that are not present in their native language" [9]. More specifically, we assume that "the initial state of the learner's perception system is a copy of her L1 perception system" [5].

Second language learners perform a process called *category assimilation*, two frequent patterns of which are *single-category assimilation* and *two-category assimilation* (see Figure 1) [5]. In single-category assimilation, learners associates "a binary contrast in the L2 with only one segment in their L1" which means that they must "somehow split the category to which the contrast has been mapped". In two-category assimilation, "the learner associates a binary contrast in the L2 with a binary contrast in her L1".The latter "can cause a perceptual problem, namely a boundary mismatch in the learner's L2 perception system, leading to problems with lexical access" [5].

It has been found that "adults can use the statistical distribution of the sounds produced in a language to infer the language's phonetic category structure" even if they are "never explicitly informed about the number of phonetic contrasts in the language" [14]. Furthermore, [9] have found that supervision can hamper the learning of phonetic categories. Consequently, we take an unsupervised approach, in which the learners are trained on the distribution of the L2 contrast pair, but are never told which category the sound belongs to.

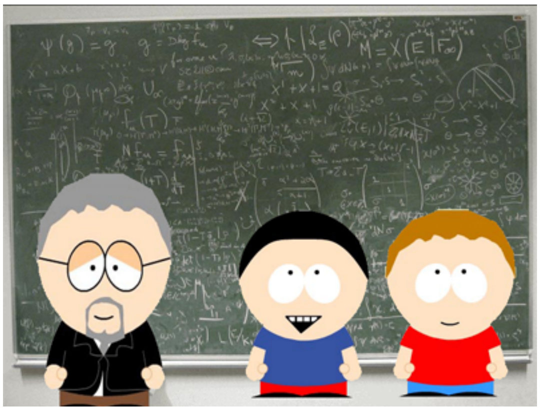


Figure 2: Screen shot of the training tool during the testing phase.

2 Experiments

Our aim was to reproduce the results obtained by [9] in unsupervised distributional learning, but with a different vowel contrast and second language learners of different language background (namely Dutch and German). As contrast, we selected the æ and ɛ of Canadian English, (example words: “cattle” and “kettle”), as this is a well-know source of difficulty for both our target groups.

We built a Flash-tool for the experiments, which allowed us to run our subjects through a pre-test, a training phase and a post-test. The testing takes place in a “classroom setup”, where two virtual students try to imitate a sound produced by their virtual teacher and the subject is asked to select the student who does it best (see Figure 2). As the students always produce prototypes of the vowel contrast, subjects thus perform a classification task of the teacher’s sound. The assignment of the prototypes to the individual students is done randomly. Likewise, the order in which the items are presented to the subject is automatically randomized. During the training phase, the subject is asked to play an ARKANOID game for distraction. During the game, whenever the ball hits the bat, the next training sound is played (in randomized order). As there is no indication of the kind of the sound category, the training is unsupervised. The game stops when all sounds have been played. The tool can flexibly be adjusted to different contrast, training or testing styles for future re-use via an xml-configuration file.

For the training of each of the contrastive pairs, we used a continuum on the F1 dimension consisting of 8 different stimuli. Stimuli 2 and 7 correspond to the prototypical /ɛ/ and /æ/, respectively. For the F2 dimension, we used a constant value of 1575 which we found to be equidistant of the male averages for /ɛ/ and /æ/ in our stimuli. These 128 sounds were played to the subjects with a frequency distribution as in [9] (see Figure 3).

For the testing, we used natural stimuli consisting of vowels in context (i.e. “beck”

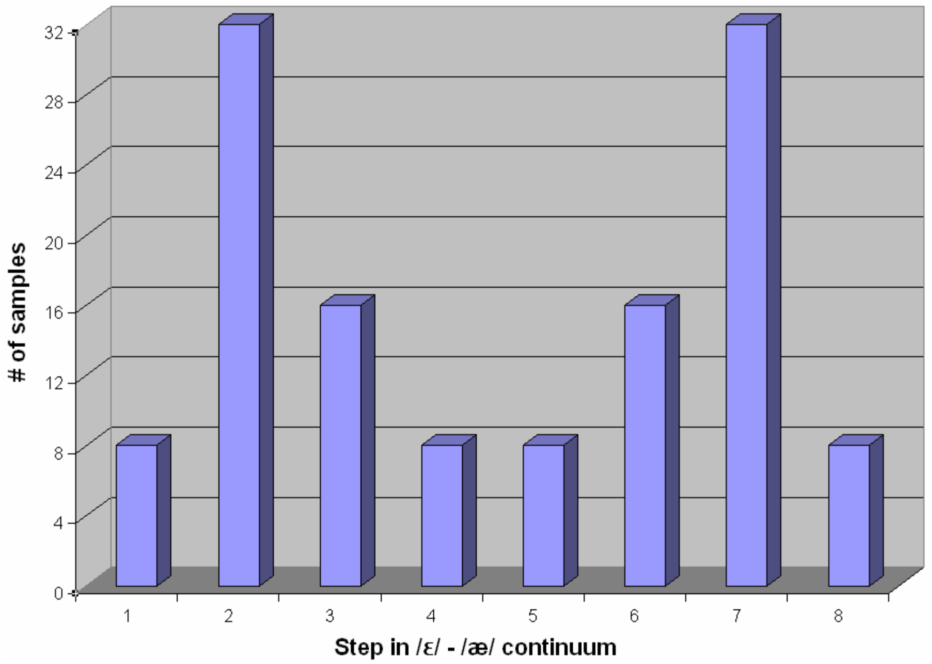


Figure 3: The distribution of the training stimuli (adapted from [9]).

/ "back" and "vest" / "vast") by 3 male and 3 female speakers (see Figure 4). For the prototypes, we choose the speaker whose F1 values in the relevant stimuli were closest to the average of all our speakers in these stimuli (male speaker).

3 Simulation

In contrast to the experiments with humans described in Section 2, the simulation consists of two phases. During the first phase, we simulated the acquisition of the native language of our subjects. Hereby, we limited the training to those vowels in our learner's native perception system, which we considered as relevant for the classification of the new vowels. Thus, we trained the simulated Dutch learners of English on the vowels /ɛ/ (as in 'bed') with F1=583Hz and /a/ (as in "zaal") with F1=795 [16]. Similarly, the German learners were trained on the vowels /ɛ/ with F1=486 (as in "Bett") and /a/ with F1=674 (as in "Rad") [18]. This leaves it open if the new contrast will be learned by single- or two-category assimilation (cf. Section 1).

During the second phase, where our simulated learners were exposed to the new contrast, we attempted to stay as close to the experimental setup with the human learners, as possible. The same holds for the testing phase, which includes male as well as female testing stimuli. While human subjects are generally able to make a generalization for vowel classification in female voices (see Section 2), we

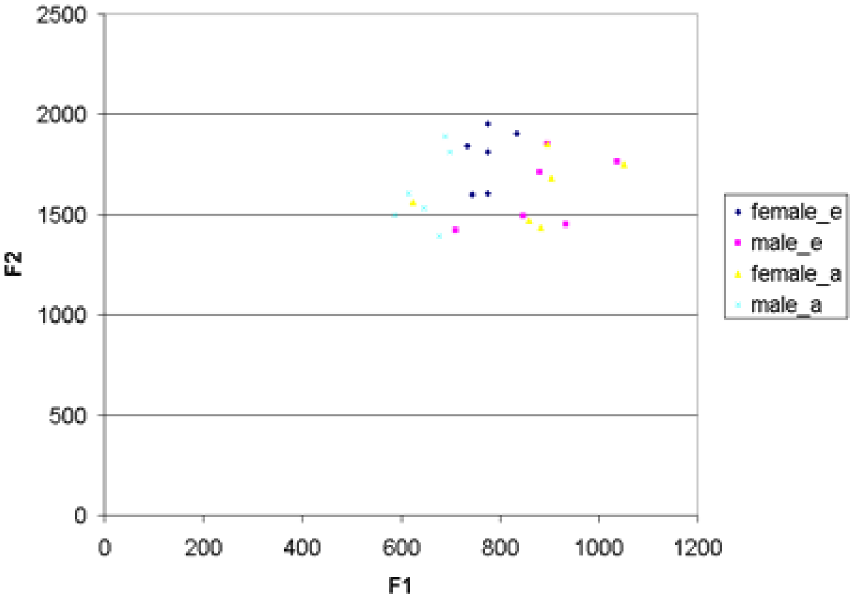


Figure 4: The natural stimuli used for pre- and post-testing throughout the experiments.

cannot expect this from the machine learning algorithms. For this reason, we have also created a gender-normalized version of the test stimuli. For this we calculated the ratio between the male and female F1 values, and then multiplied the female F1 values by this ratio to make them more male-like.

Apart from being unsupervised, there was another important requirement for the selection of the learning algorithms: It has to be able to discover the number of classes (clusters) by itself. For this reason, typically used clustering algorithms, such as k-means could not be used, as they require the number of classes specified in advance. We aimed at having a variety of very different learning algorithms. As a first step, we used a Kohonen self-organizing map to visualize the structure of our data, which is presented in Section 3.1. Next, we decided to have one well-established learning algorithm and compare its result to a (possibly new) learning algorithm we developed ourselves. As the former, we chose an unsupervised Optimality Theory (OT) learning algorithm, which has successfully been used for a vowel classification task in [2]. This is presented in Section 3.2. The newly created algorithm, which we call Time Driven Kernel Estimator (TDKE) is presented in Section 3.3.

3.1 Kohonen Self-Organizing Map

Self-organizing maps (SOM) were developed by Prof. Teuvo Kohonen in the early 1980s and can now be considered as “one of the most popular artificial neural

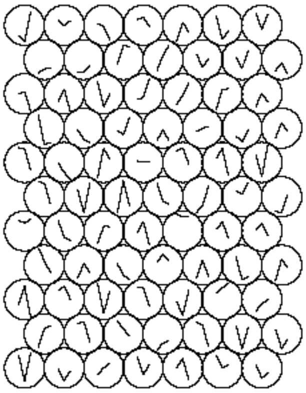


Figure 5: Schematic representation of a Kohonen self-organizing map [10]. The vectors of the nodes are initialized at random.

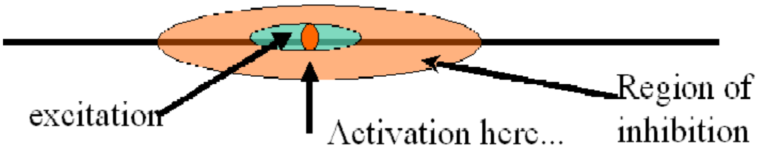


Figure 6: The principle of lateral inhibition: As a node is activated, its close neighbors get exited while more remote neighbors are inhibited. Nodes outside this range are not affected.

network algorithms" [10]. It is an "effective software tool" for the visualization and abstraction of (high-dimensional) data, as the network is "automatically organized into a meaningful two-dimensional order", with similar nodes closer to each other than dissimilar ones [12].

As shown in Figure 5, a SOM consists of a "sheet-like neural network array" [10]. Each of the cells (or nodes) i that constitute the map, are equipped with a model vector $m_i(t)$, which during training "become specifically tuned to various input signal patterns or classes of patterns in an orderly fashion" [10]. In order to achieve this, the following unsupervised competitive learning process of two steps is executed for each sample t: First, the sample's input vector $x_i(t)$ is compared to all the model vectors and the *winner* $m_c(t)$ is determined as the most similar vector in the map. Then, $m_c(t)$ as well as a number of its neighboring nodes are changed such that they come closer to the input vector.

At first glance, this activation of neighboring nodes strongly reminds of a different neural network, called *lateral inhibition*. As depicted in Figure 6, in lateral inhibition very close neighbors are excited while neighbors in a wider environment are inhibited and cells further away are not affected. Thus, in lateral inhibition, a neuron also inhibits its neighbors to determine more precisely the origin of a stimulus (contrast enhancement) [1]. In contrast, the effect of neighborhood activation in a SOM is that of “smoothing or blurring kernel over the grid”[12]. Since lateral inhibition “could be responsible for self-organization in biological systems” attempts have been made to incorporate lateral inhibition into a SOM, for a “biologically more plausible implementation” [11].

However, SOM can be considered as “a clustering method closely related to the k-means” [15]. In k-means, each point of the data set is associated with one of k centroids. Then, similarly to the adjustment of the model vectors in a SOM, the centroids are re-calculated. However, in contrast to SOM, the initial number and placement of the k centroids is crucial.

Self-organizing Kohonen maps do not require that the number of clusters is known in advance. Moreover, as the SOM is “closely modeled after neurobiological structures” [15] it “captures some of the fundamental processing principles of the brain.”[10]. First applied to text-to-speech transformation “self-organizing networks have been useful for characterizing the mechanism underlying various language acquisition tasks, and for solving statistical pattern recognition problems” which makes them “ideal for modeling the perceptual learning of phonetic categories”[8]. Most importantly, self-organizing maps have already been used successfully for vowel classification [17] and even for the classification of tones in Mandarin [8].

3.1.1 Description of the Algorithms

We used the “SOM Toolbox” (from www.cis.hut.fi/projects/somtoolbox/), which is a Matlab library based on the freeware SOM program package “SOM PAK” [13]. As described in [12], the toolbox offers two different learning algorithms, one being incremental and the other a batch version.

In the *incremental* (or sequential) learning algorithm, updates the model vectors according to

$$m_i(t+1) = m_i(t) + h_{c(x),i}(x(t) - m_i(t))$$

after each step t (i.e. after each presentation of a new input vector $x(t)$). The *neighborhood function* $h_{c(x),i}$ is chosen such that the number of neighboring nodes is initially “fairly large but it is made to shrink during learning”. This is meant to “ensure that the global order is obtained” before “local corrections of the model vectors in the map will be more specific” [10].

In contrast to the incremental algorithm, the *batch* version does not change the model vectors of the map immediately after receiving a new input vector $x(t)$. Rather, all $x(t)$ are listed under their respective closest model vector m_i^*. These lists are then united with the neighbors of m_i^* to form the union U_i. Only after all $x(t)$ have been collected like this, the means of the vectors $x(t)$ in each U_i are computed, and the old values of m_i^* are replaced by the respective means. This process is repeated until a stable result is reached.

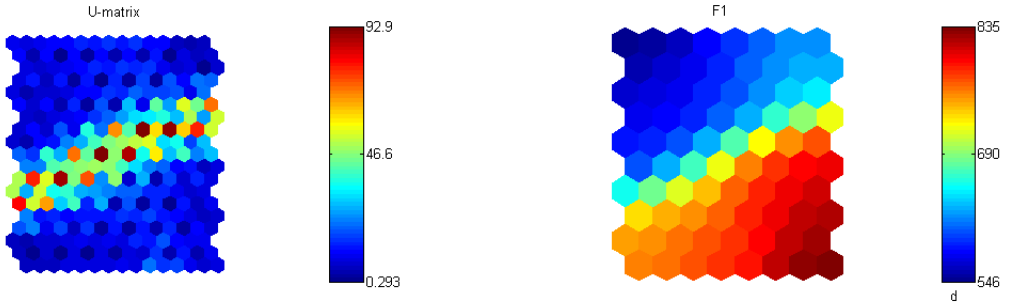


Figure 7: Kohonen simulation of an adult native speaker of Dutch.

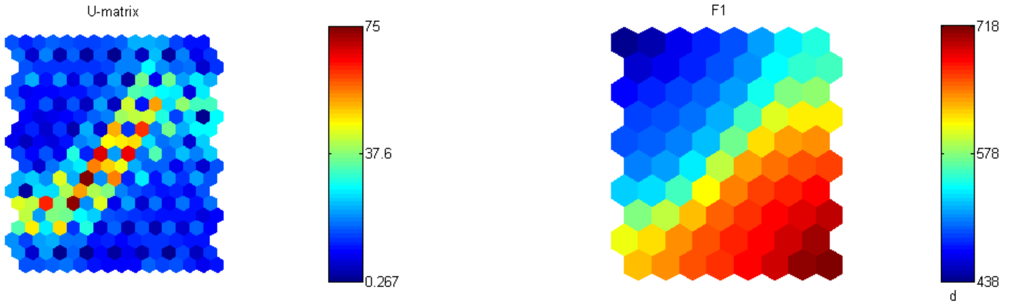


Figure 8: Kohonen simulation of an adult native speaker of German.

We are aware that the batch way of learning is more unlikely to be a close simulation of human learning than the incremental one. After all, our human subjects are only presented once with each training sample, and it is more plausible that they immediately make use of the new information rather than only incorporating it after the end of the training phase. However, the batch version "is significantly faster" than its incremental equivalent [12]. Comparing the two version therefore seems beneficial, as due to the small amount of training data the incremental method might not clearly show the desired result.

3.1.2 Result of the Simulation

As in [8] we used a Kohonen self-organizing map to "reveal the structure of the data".

For the simulation of an adult native speaker, we trained two maps with 250 /ɛ/ and 250 /a/ vowels of Dutch and German, respectively. For the F1-value, we used continuous data, produced with the standard deviation of 30. The F2-value was held constant at 1575 Hz (as in human experiments in Section 2).

The result is shown in Figures 7 and 8 for Dutch and German speakers, re-

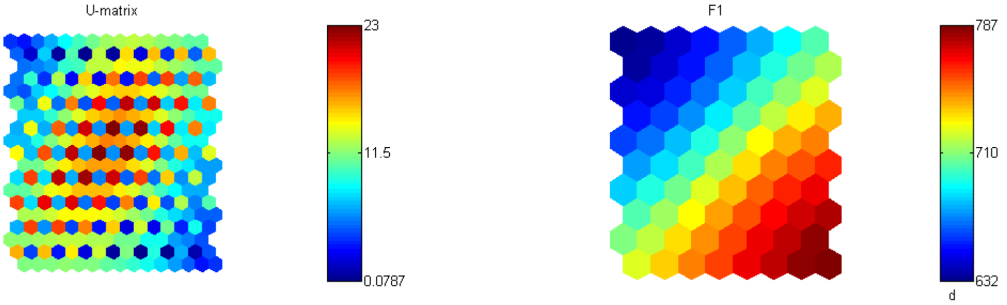


Figure 9: Sequential Kohonen simulation of a Dutch adult learning the new contrast.

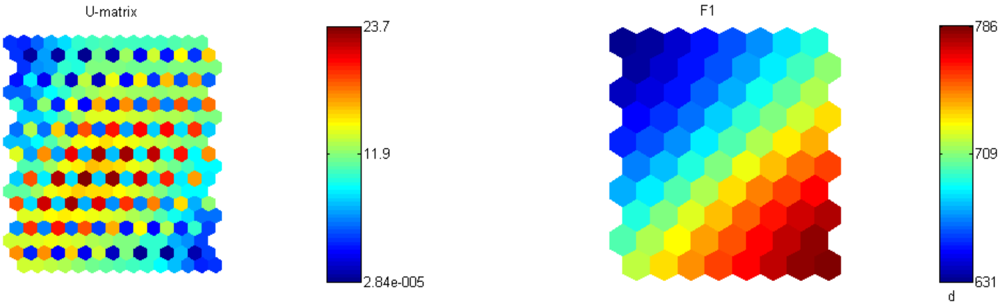


Figure 10: Sequential Kohonen simulation of a German adult learning the new contrast.

spectively. The U-matrix on the left shows the difference between the neighboring nodes, with blue representing a small, green a medium and red a large distance. Both U-matrixes indicated that the map was successful in learning the concept of two clusters. The maps on the right shows the distribution of the F1-values. According to these maps, the borders between the two categories are at approximately 690 Hz and 578 Hz for Dutch and German native speakers, respectively, which is a reasonable result. The two clusters in the map for the Dutch speakers might be slightly clearer, because the difference between the average F1 values of /ɛ/ and /a/ is bigger in this language.

For the simulation of the experiment, we used data created in exactly the same way as in the experiment in Section 2. We compared the two different modes of training the SOM: sequential and batch. As described in Section 3.1.1, the difference between the two modes is that during batch training, the map is not changed immediately, but only after the completion of a whole epoch and that in the batch version the samples might be presented several times.

As shown in Figures 9 and 10, sequential training does not result in any signif-

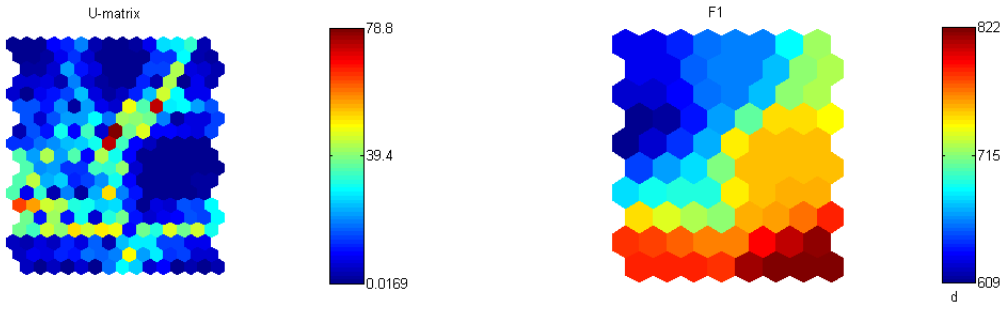


Figure 11: Batch Kohonen simulation of a Dutch learning the new contrast.

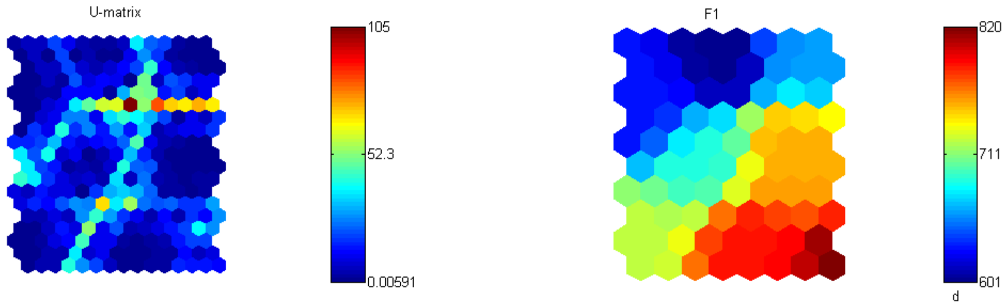


Figure 12: Batch Kohonen simulation of a German learning the new contrast.

icant changes of the F1-value representation. However, the U-matrix (which visualizes the difference of the vectors of neighboring nodes) shows that the clustering is much less clear, indicating that our simulated learner is in a state of confusion or uncertainty. After the batch training, on the other hand, new clusters seem to emerge, which are reflected in the F1 visualization as well as in the U-map (see Figures 11 and 12). However, the number of the new clusters is not as clear as in the simulation of the adult natives, as a repetition of the experiment resulted in a different shape and number of the clusters. Clearly, neither of the algorithm was able to fully grasp the new concept to be learned from the limited amount of data it was presented. While the training has clearly caused some change, its result is neither unambiguous nor stable, yet.

3.2 Optimality Theory

3.2.1 Modeling a Beginning L2 Learner

The second algorithm that we used to simulate the unsupervised training process was Stochastic Optimality Theory with the Gradual Learning Algorithm, as in [2] and [7]. To model Dutch and German speakers respectively, we used Praat [3] to train an empty F1 grammar with a distribution of input-output pairs ranging from 300 Hz to 1100 Hz, on two F1 distributions for /a/ and /ε/, in steps of 20 Hz each. The mean values for these phonemes in both languages were taken from [16] and [18] respectively, and a standard deviation of 45 Hz was assumed to make a distribution of both vowels with peaks around these values.

The Stochastic OT algorithm is not in any way predisposed toward learning a certain number of vowels. By exposing it to a large number of F1 values (vowels) that are distributed in a certain way, the algorithm learns to perceptually warp a series of inputs to a single output that corresponds with a peak in the distribution that it is trained on. Thus the range of input candidates is reduced to a smaller number of winning output candidates. For example the trained Dutch grammar has /440 Hz/ as a winning output for inputs [400 Hz], [420 Hz], [440 Hz] and [460 Hz]. The grammars of the simulated Dutch and German L1 speakers were trained on 10,000 epochs of the training data.

3.2.2 Modeling the Testing and Training

With these trained F1 grammars now representing an adult who has been exposed to the vowels of his native tongue for his whole life, we first took the pre-test: the algorithm was fed with F1 input forms corresponding to those used in the human testing tool. Since the task for humans was not to classify the vowel by itself, but to point out which 'prototype' it was closest to, we modeled this answer by checking which of the outputs corresponding to the prototypes in the test (/620 Hz/ and /840 Hz/ respectively) was most optimal; that is, we checked which one was marked with 'fatal violation' further in the OT tableau. As the OT grammar created only accepts inputs in steps of 20 Hz, we rounded the F1 values of the test samples to the nearest of the steps (e.g. an F1 of 833 Hz became an input of [840 Hz]).

Next, we exposed these trained grammars to the F1 values of the training continuum described in Section 2. Since training on only 128 tokens, as in the human learners, barely affected the grammars, we have chosen to train it on about 1,000 tokens. Figure 14 illustrates the input/output space of the grammars before and after training. We then again looked at the output values for the F1 input forms corresponding to the test sounds. Figure 13 shows the classification of the test samples by the simulated German speaker before training on the stimuli (i.e. the results of the pre-test for this simulated person). The complete results of the pre- and post-training tests are given in the next section.

3.3 A Time Driving Kernel Estimation Algorithm

Our Time Driven Kernel Estimator or TDKE algorithm, can learn vowels in an unsupervised way. Since it is sensitive for the order in which it learns samples, it is able to model shifts in concepts over time. Unlike other methods, it does not need a predefined amount of target concepts, which is not only biologically plausible, but

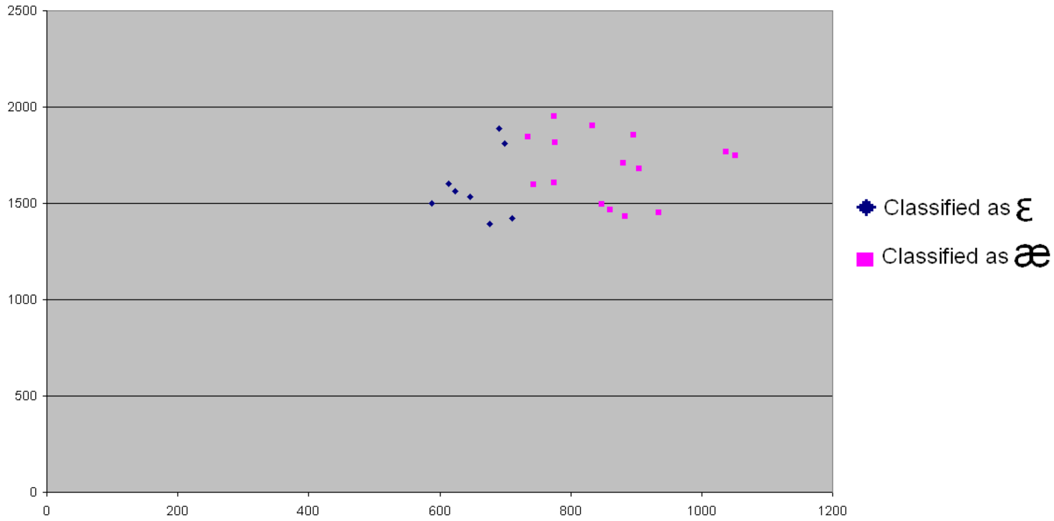


Figure 13: Classification of test vowels by simulated German OT grammar, before training on the synthetic Canadian English stimuli.

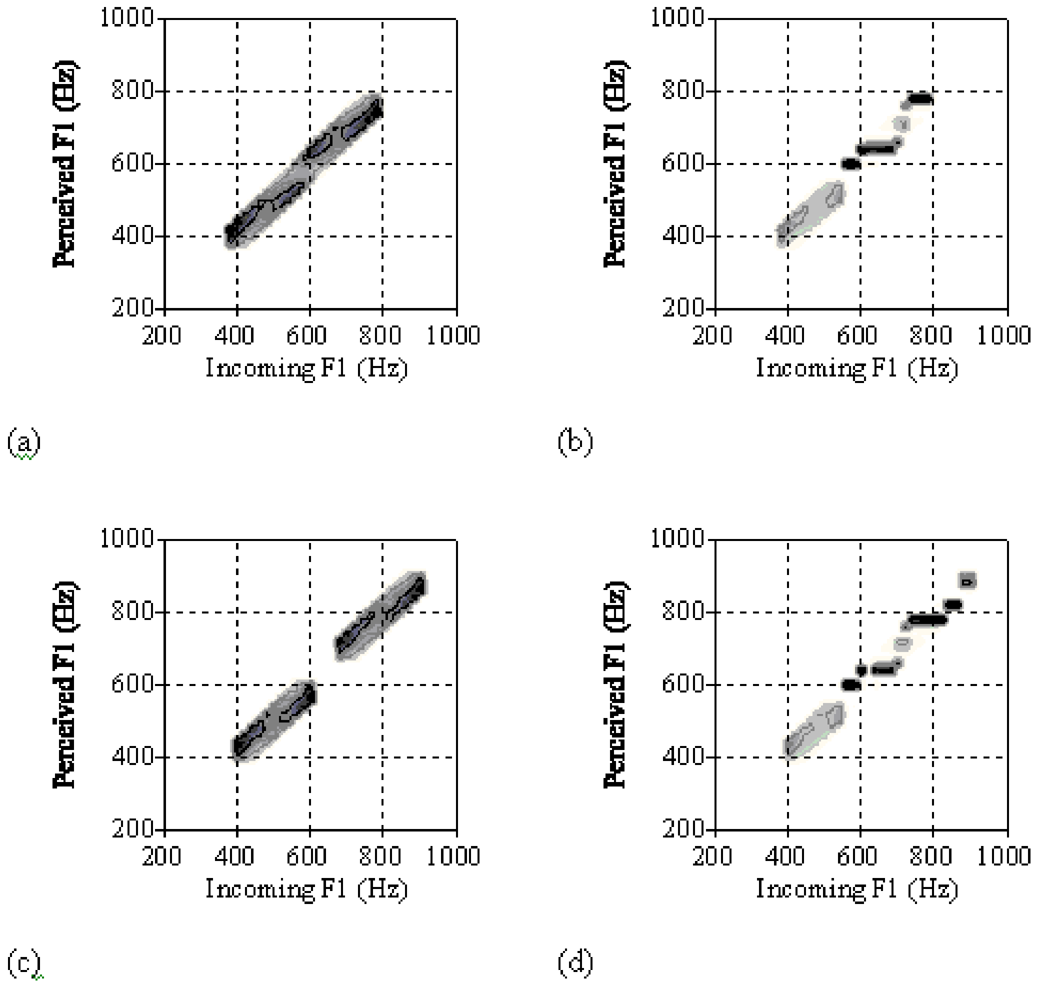
it also enables it to merge or split concepts. We adapted it to classify stochastically, since this seems to be the behavior of humans.

As mentioned in Section 1, we assume that L2 learners typically start out using the vowel abstractions from their L1 [4]. Over time these abstractions shift to the formant frequencies of the new language, in this process abstractions might split into more or merge to less [6]. The learning of these new vowels seems data driven [6],in other words: It does not happen over time, old abstractions get faded by contradicting data . Note however that these new categories don't override original ones; learners can still interpret L1. Most likely, the old categories get copied in a new model for the new context. We propose an algorithm, for learning vowel distributions, that meets this and other conditions.

3.3.1 The Mechanics

In modeling this, it seems logical to consider all learned data, but in a weighted manner. Old data does have an impact but less after new facts have been discovered. In this case we use order to determine the weight, in real life this will probably depend on a more complex notion of relevance such as context of which time is undoubtably a factor. For this memory-weight we have used a custom hockey stick graph (see Figure 15), given by:

$$\frac{1}{1+e^{-(\frac{1}{2}r-(r(1-\frac{i}{N}^{w})))}}$$



(a)

(b)

(c)

(d)

Figure 14: Warping of the perceptual space. These plots show to which output an incoming F1 is mapped in the OT analysis. A blacker area means a higher concentration of winning outputs. (a) is the modeled German speaker before training on the Canadian English stimuli, (b) the German speaker after training, and (c) and (d) show the same picture for the simulated speaker of Dutch. Note that the discrete nature of the 8-step training stimuli and their relative weight in the data make for less evenly distributed output data.

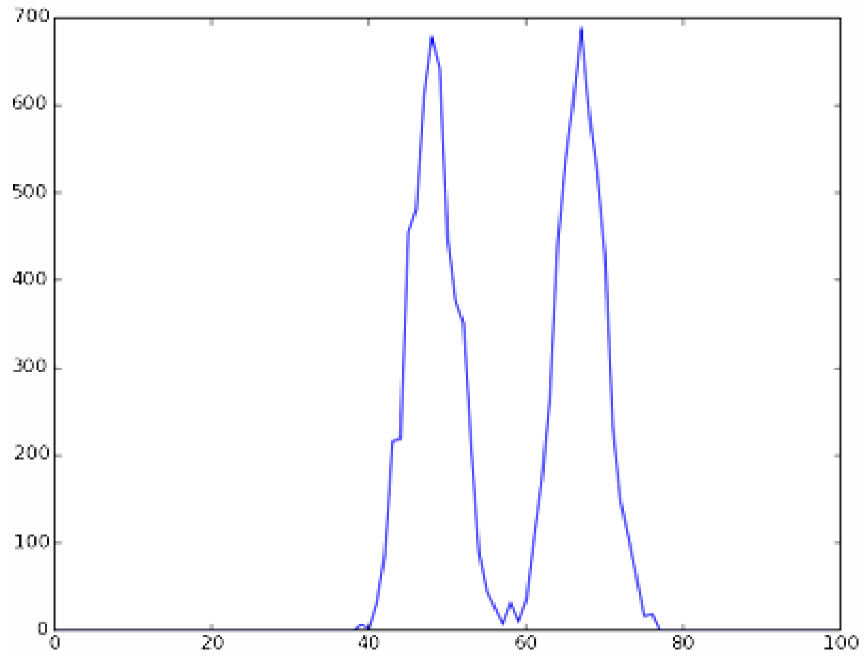


Figure 15: Histogram of a trained German speaker (two vowels), with memory weighed samples.

The basis of this function is the classic sigmoid, since new instances are all pretty relevant until a certain cut off point. The r is the range of the classic sigmoid we want the use in this experiment. We have chosen to use 10, which means that we will look at -5 to 5. The fraction i over N returns a value between 0 and 1 and expresses in that the relative age of the sample. The w is the power which warps the value of the fraction shifting the cut off point up. In this experiment we use $t = 70$ (though 6 seemed more natural, one of the requirements was that 128 samples should be enough to shift a 20000 sample base). The advantages from this function are that new examples are all relatively well considered, till a certain cutoff point after which samples hardly count. Since very old examples do still count and the list of examples never shrinks, the newest examples get less and less impact with the size of the train set. This would model loss in plasticity.

From the simulated memory (see Figure 16) we construct a model, according to the standard Parzen Window Method, in this case by fitting gaussians, which seems to be a reasonable distribution for human vowels. The guassian kernel is given by the following equation:

$$\frac{1}{\sqrt{2 * \pi}} * e^{-\frac{x^2}{2}}$$

The method by Emanuel Parzen is given by the following equation, in which K is a kernel function:

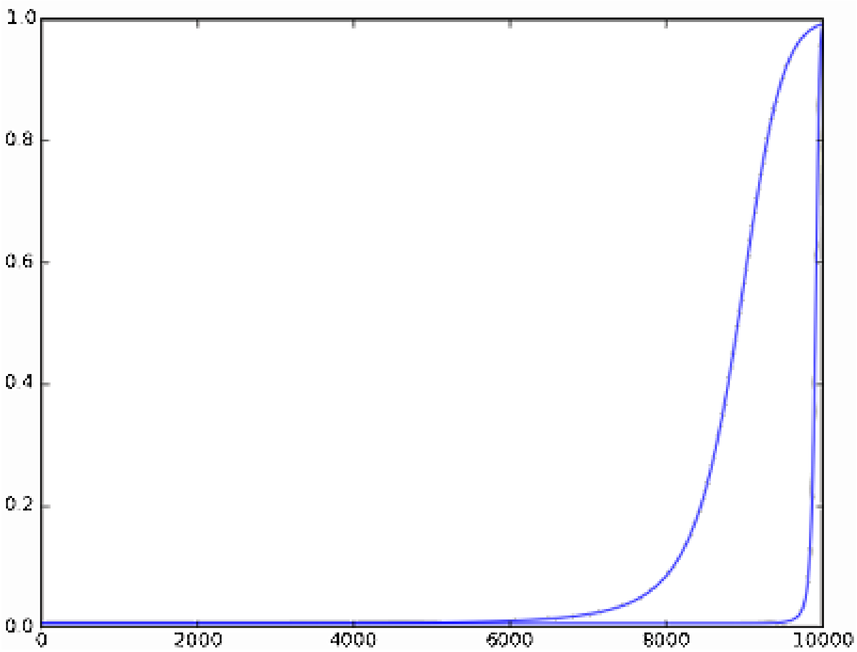


Figure 16: The resulting memory weight function for 10000 items with warp of 6 and 70.

$$\frac{1}{n * h} \sum_{i=0}^{n} K \frac{x - x_i}{h}$$

Depending on the amount of smoothing used in the method (the factor h) we will find 1 or more peaks in the model, each peak can be regarded as category. In order to classify a stimulus our model can now calculate the distance to a peak and return the closest peak as the class to which the stimulus belongs. Note however that for the time being we use absolute distance, in other words the distance in frequencies. It would be feasible to use the relative distance, by considering the deviation of a class (which is indirectly given by the height of the peak and moreover the falloff).

3.3.2 The Algorithm

The actual algorithm is designed with 5 step in mind: Sampling, Memory, Experience, Abstraction and Classification. In our particular implementation we have distinguished 3 stages: Loading Settings, Loading Functions and Actions. Each of these phases takes the five afore mentioned steps respectively.

Upon the actual Action stage, we build a sequential list of samples of L1, which we then extend with the L2 samples.

Next we build a weighted histogram by going over the entire list summing the memory weights (each occurrence is not just simply counted as 1 but as weighted value expressing it's contextual relevance or in this case recentness).

From the weighted histogram (see Figure 16) a series of samples get constructed, by multiplying the values by 10000 and rounding them off each count is now considered as a sample. We could regard this set as the set of samples a speaker would reproduce from memory (though it is quite extensive).

Now, from this set kernels get estimated in a rather straightforward way, by just fitting kernels on every bin, and combining these given a certain smoothing parameter. This is the actual implementation of the mentioned Parzen Method.

The values of this function per bin is used to construct the estimated distribution in the following manner:

We can now assume that every maximum in our kernel estimation is a vowel category, after all these are the estimated distributions. Since we assumed all standard deviations to be equal, we did so at the start when building the test and training samples, but also implicitly at the kernel estimation part, we can now simply express relatedness to a category by counting the number of bins between sample and concept. From this distances we pick the shortest, give or take some stochastic noise.

Note that with slight adaptation we could give the algorithm a notion of different deviations, which would be similar to the warping of perception in OT (peaks with many samples under them would be broader and thus the relative distances smaller).

3.3.3 The Characteristics

This model does not assume a given number of categories and smart smoothing choices make pruning obsolete. This sets it apart from methods such as K-means. Because the histograms can be extended infinitely without significantly hurting the performance we are not bound to ranges as we would be with connectionist approaches. As of yet we do not use relative distances as in OT or some other methods, this does not seem to hurt performance, but this deserves investigation. The method is stochastic but its firm mathematical nature makes it easier to analyze and adapt than more rule bound methods such as OT. This nature also makes it very adaptable for more dimensional data or completely different ranges.

3.3.4 The Results

The results for German speakers are shown in Figures 17 and 18. Preliminary, we would like to mention that the requirement of using only 128 retrain samples was tough, we had to tweak quite a lot for that. Secondly we would like to add that just considering the F1 meant that some samples were just doomed to be misclassified, they we just too far out. It is very obvious that the algorithm would have

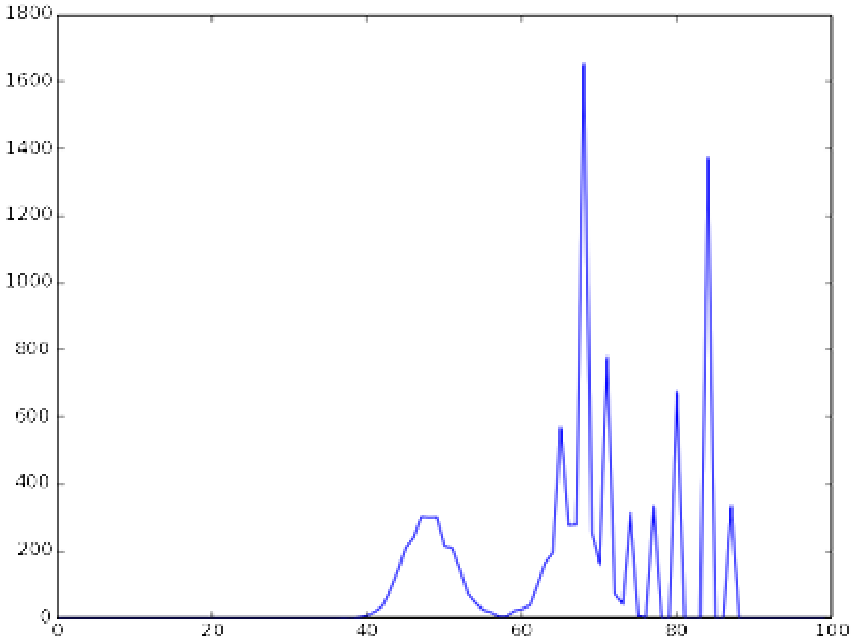


Figure 17: Histogram of a trained German speaker retrained with 128 new samples of Canadian English.

performed far better if it was allowed to look at the F2 as second dimension or a linear combination of F1 and F2.

One of the things we could still implement, is considering the height of a bump in classifying, in other words the a priori chance of a category, something we have not done yet. Looking at the results the /ɛ/ is the harder concept to learn, about half of those get misclassified, while the /æ/ gets classified correctly almost all the time. This would contradict with the helpfulness of the a priori change (or the height of the bump) since the /æ/ is the larger category (it coincides with the old /æ/ more).

It seemed that the program had trouble distinguishing the F1's on account of the large variations caused by intra-speaker variation. Some of these variations caused by factors such as gender differences are strongly correlated to ignored dimensions, in this case F0 for instance might help, though F2 might be sufficient.

The resulting classification seemed to favor the /æ/-label as shown in Figure 19.

4 Results and Analysis

As described in the previous sections, we exposed three human subjects (two L1 speakers of Dutch and one L1 speaker of Standard German) to 128 synthetic vowel

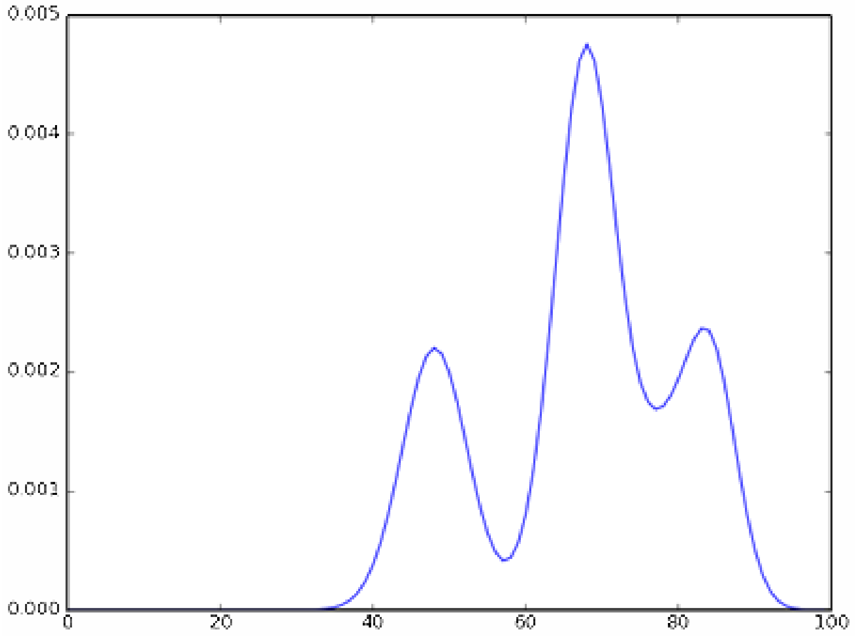


Figure 18: Estimation of vowel distribution in a modeled German speaker that was retrained in Canadian. On the Left is the old German E in the middle the new Canadian E.

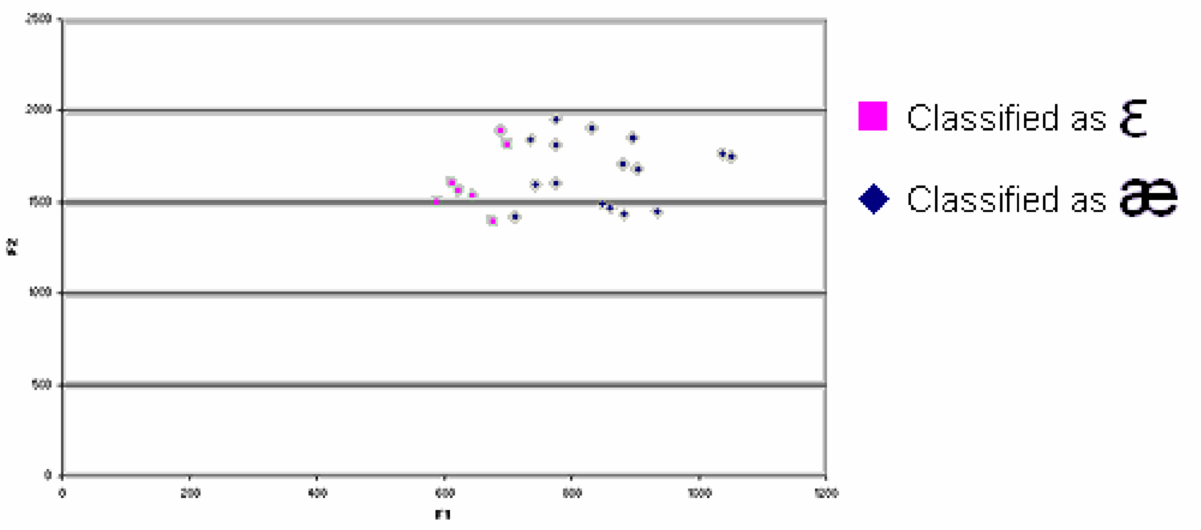


Figure 19: The resulting classification of the TDKE-model trained in Dutch (retrained 128 times).

stimuli from an 8-step continuum from /ɛ/ to /æ/, and took a discrimination test before and after the training. We also trained three learning algorithms on the same data to simulate the human learning process. For two of these (Stochastic OT and the TDKE) we also simulated the discrimination test; with the Self-Organizing Maps this was not possible, but the maps themselves did give some insight into way this learning method learns to categorize the vowel data. Since the batch version performs somewhat better than the sequential algorithm, we conclude that the desired information about the new categories could be extracted from the training data, but the amount of training data presented to the SOM is not sufficient to allow for a stable result.

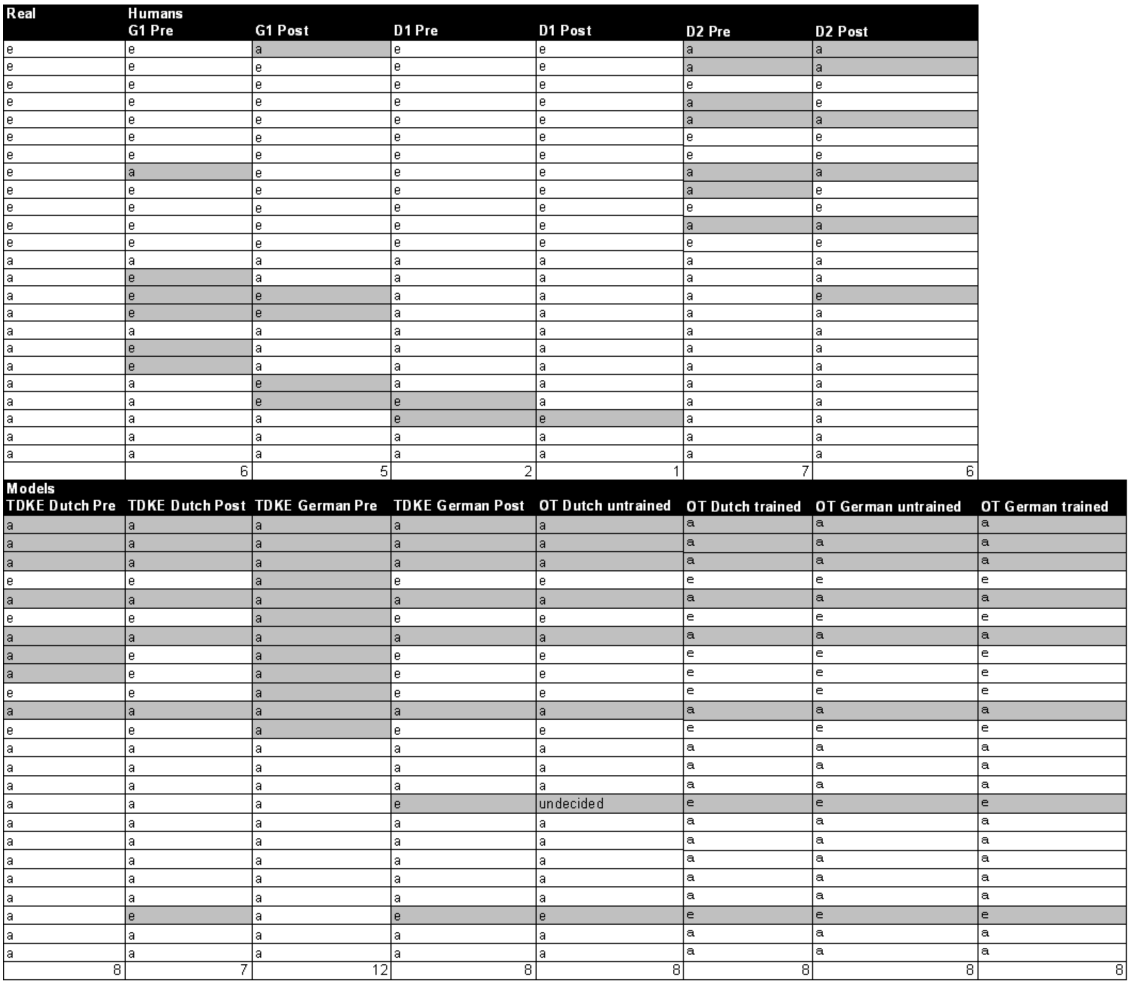
Figure 20 shows the outcome of the prototype identification task for the humans and the two algorithms. The human learners all seem to show a slight improvement between the pre- and post test. However, due to the small number of test subjects it is not possible to say whether the training accounts for this. Interestingly, the human learners vary not only in the amount of errors made in classification, but also in the nature of these errors. German speaker G1 and Dutch speaker D1 both make (almost) all of their misclassification by labeling /æ/ as /ɛ/. Dutch speaker D2, on the other hand, mostly mis classifies in the other direction, wrongly identifying /ɛ/ as /æ/. These variations between speakers are interesting, since single-category assimilation seems to take place for all speakers, but sometimes in opposite directions. Again however the small number of subjects does not warrant making any concrete statements about the cause of this variation.

The TDKE algorithm shows a slight improvement after training in the case of the simulated Dutch speaker, and a great improvement in the case of the simulated German speaker, which at first classifies all F1 values as /æ/. The Stochastic OT on the other hand shows no improvement at all after training. Note that the TDKE was tested on non-normalized data only [1], while OT was tested on both non-normalized and gender normalized data (see Section 3). However gender normalization did not have any effect on the classification result.

5 Conclusions and Future Work

In this project we have looked into the effects of unsupervised learning of bimodally distributed vowel data, with both real and simulated language learners, on the perception of a new vowel category. For this we developed an unsupervised learning and testing tool which is easy to understand for subjects, and can be freely adapted to train and test different sounds for future research. Preliminary results of the tests on humans seem to suggest that there is some improvement after training, as was the case in [9] but due to the small scope of the tests this remains only a suggestion. We also presented a novel learning algorithm, the Time-Driven Kernel Estimator, which seems to improve after training on the F1 values vowel data.

[1] Unfortunately, we lost TDKE in a harwarde crash.



Real	Humans					
	G1 Pre	G1 Post	D1 Pre	D1 Post	D2 Pre	D2 Post
e	e	a	e	e	a	a
e	e	e	e	e	a	a
e	e	e	e	e	e	e
e	e	e	e	e	a	e
e	e	e	e	e	a	a
e	e	e	e	e	e	e
e	e	e	e	e	e	e
e	a	e	e	e	a	a
e	e	e	e	e	a	e
e	e	e	e	e	e	e
e	e	e	e	e	a	a
e	e	e	e	e	e	e
a	a	a	a	a	a	a
a	e	a	a	a	a	a
a	e	e	a	a	a	e
a	e	e	a	a	a	a
a	a	a	a	a	a	a
a	e	a	a	a	a	a
a	e	a	a	a	a	a
a	a	e	a	a	a	a
a	a	e	e	a	a	a
a	a	a	e	e	a	a
a	a	a	a	a	a	a
a	a	a	a	a	a	a
	6	5	2	1	7	6

Models							
TDKE Dutch Pre	TDKE Dutch Post	TDKE German Pre	TDKE German Post	OT Dutch untrained	OT Dutch trained	OT German untrained	OT German trained
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
e	e	a	e	e	e	e	e
a	a	a	a	a	a	a	a
e	e	a	e	e	e	e	e
a	a	a	a	a	a	a	a
a	e	a	e	e	e	e	e
a	e	a	e	e	e	e	e
e	e	a	e	e	e	e	e
a	a	a	a	a	a	a	a
e	e	a	e	e	e	e	e
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	e	undecided	e	e	e
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
a	e	a	e	e	e	e	e
a	a	a	a	a	a	a	a
a	a	a	a	a	a	a	a
8	7	12	8	8	8	8	8

Figure 20: This table shows the results of the pre- and post-test classification task as performed by the human learners and the two learning algorithms. G1 is the native speaker of German and D1 and D2 are native speakers of Dutch. The first column 'Real' shows the the actual vowel that was produced (either 'e' as in 'beck'/'vest' or 'a' as in 'back'/'vast'), and the other columns show to which prototypes the test subjects and algorithms matched this sound. Erroneous classifications are marked in gray.

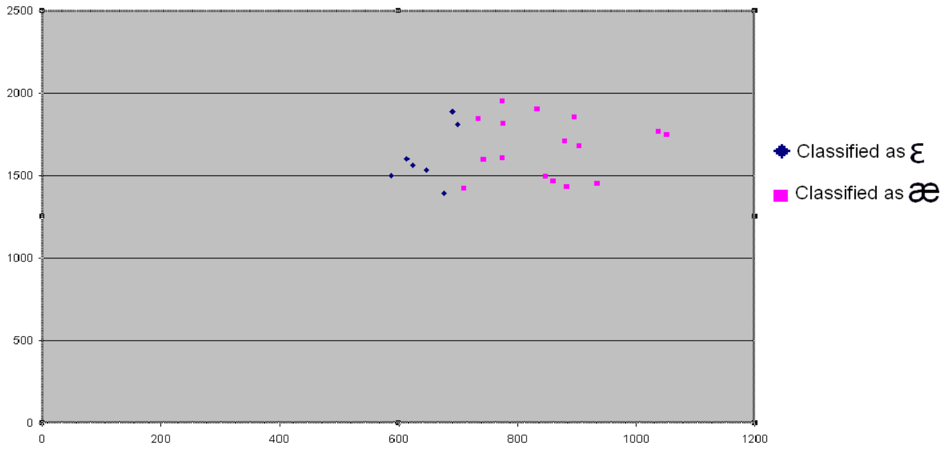


Figure 21: Resulting classification of the OT-model trained in Dutch (re-trained 1000 times).

The other algorithm used, Stochastic Optimality Theory with the Gradual Learning Algorithm, showed no improvement after training.

As future work we propose to make slight changes to the training tool, such that the game will be more exiting and demanding for the subjects (in particular, by adding more difficult levels for subjects with prior expertise in the game). This is because we had the impression that our subjects might not have paid enough attention to the sounds as they were bored by the game. When this has been accomplished, the tool should be ready to be used on a larger number of subjects.

References

[1] Pradeep Arkachar and Meghanad D. Wagh. Criticality of lateral inhibition for edge enhancement in neural systems. *Neurocomput.*, 70(4-6):991–999, 2007.

[2] Paul Boersma, Paola Escudero, and Rachel Hayes. Learning abstract phonological from auditory phonetic categories: An integrated model for the acquisition of language-specific sound categories. In *Proceedings of the 15th International Congress of Phonetic Sciences*, pages 1013–1016, Barcelona, August 3–9 2003.

[3] Paul Boersma and David Weenink. *Praat: doing phonetics by computer (Version 4.6.09)*, 2007. Retrieved June 24, 2007, from http://www.praat.org/.

[4] Rex A. Sprouse Bonnie D. Schwartz. L2 cognitive states and the full transfer/full access model. *Second Language Research*, 12:40–72, January 1996.

[5] Paola Escudero and Paul Boersma. The subset problem in l2 perceptual development: Multiple-category assimilation by dutch learners of spanish. In *Proceedings of the 26th Boston University Conference on Language Development*, 2002.

[6] Paola Escudero and Paul Boersma. The subset problem in l2 perceptual development: Multiple-category assimilation by dutch learners of spanish. In Barbora Skarabela, Sarah Fish, and Anna H.-J. Do, editors, *Proceedings of the 26th Annual Boston University Conference on Language Development*, pages 208–219, 2002.

[7] Paola Escudero and Paul Boersma. Modelling the perceptual development of phonological contrasts with optimality theory and the gradual learning algorithm. In *Proceedings of the 25th Penn Linguistics Colloquium. Penn Working Papers in Linguistics*, 2003. Also in ROA-439, Rutgers Optimality Archive, http://roa.rutgers.edu/.

[8] Bruno Gauthier and Rushen Shi. Simulating the acquisition of lexical tones from continuous dynamic input. *The Journal of the Acoustical Society of America*, 121(5):EL190–EL195, May 2007.

[9] Margarita Gulian, Paola Escudero, and Paul Boersma. Supervision hampers distributional learning of vowel contrasts. In *Proceedings of the International Congress of Phonetic Sciences*, Saarbrücken, 2007.

[10] Timo Honkela. *Self-Organizing Maps in Natural Language Processing*. PhD thesis, Helsinki University of Technology, Finland, December 1997.

[11] T. Kohonen, K. Makisara, Olli Simula, and Jari Kangas. Self-organizing process based on lateral inhibition and synaptic resource redistribution. In

Proceedings of the International Conference on Artificial Neural Networks (ICANN-91, Espoo, Finland), pages 415–420, 1991.

[12] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21:1–6, 1998.

[13] Teuvo Kohonen, Jussi Hynninen, Jari Kangas, and Jorma Laaksonen. Som pak: The self-organizing map program package. Technical Report A31, Helsinki University of Technology, Faculty of Information Technology, Laboratory of Computer and Information Science, January 1996.

[14] Jessica Maye and LouAnn Gerken. Learning phonemes: How far can the input take us? In *Proceedings of the 25th Annual Boston University Conference on Language Development*, 2001.

[15] F. Murtagh. Interpreting the kohonen self-organizing feature map using contiguity-constrained clustering. *Pattern Recogn. Lett.*, 16(4):399–408, 1995.

[16] L.C.W. Pols, H.R.C. Tromp, and R. Plomp. Frequency analysis of dutch vowels from 50 male speakers. *The journal of the Acoustical Society of America*, 53:1093–1101, 1973.

[17] D. J. M. Weenink. Vowel classification with neural nets: A comparison of cost functions. *Proceedings of the Institute of Phonetic Sciences University of Amsterdam*, 17:1–11, 1993.

[18] M.-B Wesenick and Chr. Draxler. Phonetic analysis of vowel segments in the phondat database of spoken german. In *Proceedings of the International Conference of Phonetic Sciences*, pages 416–419, Stockholm, Schweden, 1995.