# *Term paper for Acoustic Phonetics WS20/21*
# *Wang and Lee (2015) Summary and Terminology*

*Jingwen LI and Kuan TANG* [1]

[1] Seminar für Sprachwissenschaft, Eberhard Karls Universität Tübingen
*jingwen.li/kuan.tang@student.uni-tuebingen.de*

*28. Mar 2021*

This term paper aims to introduce the main concepts and methods of Wang and Lee (2015). The content and authorship of this paper as follows: In the first section, Kuan summaries the whole paper in general, and makes clear that the research questions, methodology, and experimental results in Wang and Lee (2015). In section 2, Jingwen introduces the theoretical frameworks for both Supervised EP detection and Unsupervised EP discovery respectively as well as simple and vivid examples to explain the core model structures used in this paper. On this basis, Kuan gives more details about the concepts of clustering and metrics and the applications of both in the paper in the end.

## *1. Summary*

How to apply Speech Processing Technology in Second Language Learning (SLL)? How to generate informative feedback for Second Language Learners (SLLs) to improve their pronunciation based on Computer-assisted language learning (CALL) and Computer-aided pronunciation training (CAPT)[2]? How to offer SLLs not only quantitative measures of language proficiency but also specific types of errors they have made? These are three trends that have led to substantial efforts toward CALL to meet the strong demand of SLL. At least two different but closely related Pronunciation error patterns (EPs)[3] tasks in CAPT: First, to derive the EP dictionary for a given L1-L2 pair, or a given L2 but non-specific L1; Second, to verify whether a voice segment produced by a learner is correct, or if it belongs to a specific EP based on the EP dictionary.

[2] Computer-aided pronunciation training (CAPT) aims to analyze the produced utterance to offer feedback to the language learner in the form of quantitative or qualitative evaluations of the pronunciation proficiency.

[3] Pronunciation error patterns (EPs) are patterns of mispronunciation frequently produced by language learners and are usually different for different pairs of target and native languages.

Wang and Lee (2015) propose two novel frameworks for both Supervised Error Patterns Detection and Unsupervised EP Discovery. For supervised EP detection, they use hierarchical multi-layer perceptrons (MLPs) as the EP classifiers to be integrated with the baseline using HMM/GMM in a two-pass Viterbi decoding architecture. As for unsupervised EP discovery, they use the Hierarchical Agglomerative Clustering (HAC) algorithm to explore sub-segmental variation within phoneme segments and produce fixed-length segment-level feature vectors to distinguish different EPs.

They tested K-means [4] and the Gaussian mixture model with the minimum description length principle [5] for EP discovery. And they also propose to use the universal phoneme posteriorgram (UPP),

[4] Assuming a known number of EPs

[5] Estimating an unknown number of EPs

derived from an MLP trained on corpora of mixed languages, as frame-level features in both supervised detection and unsupervised discovery of EPs.

Preliminary experiments offered very encouraging results. As the experimental results have shown, comparing with other models, the new framework enhances the power of EP diagnosis, and using UPP achieves the best performance and is useful in analyzing the mispronunciation produced by language learners.

## 2. *Theoretical Framework*

### *2.1 Error Patterns*

**Error patterns** (EPs) in pronunciation training are frequently encountered mispronunciations that have intrinsic commonalities. In other words, mispronunciations of the same error pattern usually display a high level of intra-group similarity, although they can still be very close to the canonical pronunciation or other EPs.

EPs can help give more informative and targeted feedbacks in pronunciation training. For example, language learners of different L1s may all pronounce the word wrong (wrong as in deviation from the canonical pronunciation), but their mispronunciations can be incorrect in different ways. It would be more helpful if the CAPT system can provide feedbacks that are specific enough to show the area of improvement. EPs answer the question *"**In what way is the given pronunciation wrong?**"* instead of simply stating it as an incorrect pronunciation.

#### *2.1.1 Supervised EP Detection and Unsupervised EP Discovery*

**Supervised EP detection**, the most commonly used technique, relies on linguistic expertise of not only the target language but also the L1s of the learners. Thus defining a full-coverage EP dictionary for different language pairs or mixed languages can be very expensive. As explained above, EPs are similar to their canonical pronunciations. Apart from that, EPs that deviate from the same standard native pronunciation are also similar to each other, making EP detection a rather difficult task even if EP dictionaries can be provided, either generated manually or automatically derived (the task of **unsupervised EP discovery**). Unsupervised EP discovery aims to derive EPs from corpora directly, without language-based knowledge as a precursor.

To put it simply, supervised EP detection is like putting pronunciation instances into a divided space, like the box displayed in Figure

1, each cell is an EP advised by experts. The division of the space depends on the training of the model. On the other hand, unsupervised EP discovery does not presuppose the boundaries or the number of EPs[6] (at first). Take **Hierarchical Agglomerative Clustering** (HAC) as an example, the HAC algorithm starts with each instance in its cluster and then simply compares the instances using distance[7] between them in high-dimensional space, the most similar instances are grouped, until all instances are in the same group. Categories/clusters are in this sense "self-organized" in a bottom-up manner and can be adjusted depending on the desired granularity. This is where humans make decisions, a high cut-off at the resulting dendrogram in Figure 2 means fewer classes while a low cut-off produces more classes.

[6] except K-means

[7] There are many distance measures to choose from, to name a few, Euclidean distance, Manhattan distance...

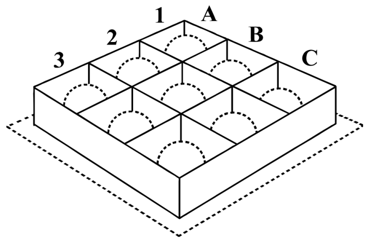

Figure 1: A divided box, adopted from p.9 of `http://fanfoyan.com/audio/naga.pdf`

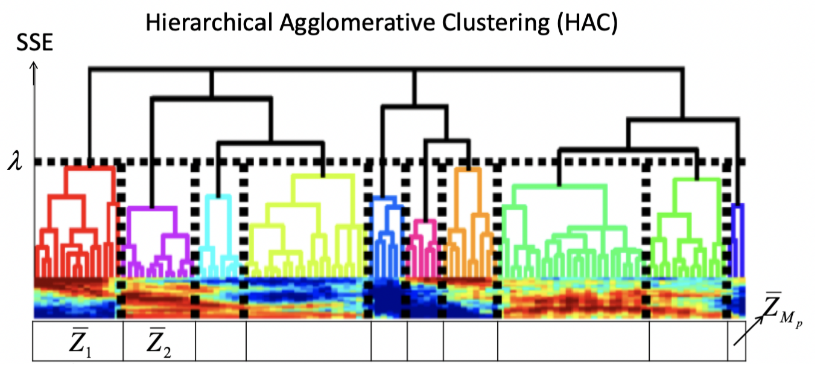

Figure 2: Dendrogram resulting from HAC, adopted from Wang and Lee (2015)

## 2.2 *Heterogeneous Initialization of Acoustic Modeling of EPs*

In supervised detection of EPs, an **acoustic model** (AM) is needed for each EP. Wang and Lee (2015) asked language teachers to create an extensive set of 152 EPs, whose descriptions are based on phonemes of Chinese and English. For example, *l_000* represents the canonical pronunciation of l, while *l_010* is defined as something pronounced like the English *l*. Since EPs are defined using a ***"pronounced-as"*** relationship[8], it became natural to use existing acoustic models for phonemes as a starting point. Duplicates of phoneme models trained for other corpora are used as the initial EP models, this process is referred to as ***heterogeneous model initialization***. A comparison of *heterogeneous model initialization* and ***homogeneous model initializa-***

[8] Notice that the ***"pronounced-as"*** relationship only describes a pronunciation instance and has nothing to do with the learners' L1.

***tion*** is illustrated in Figure 3, where a clear benefit of heterogenous initialization can be seen: Without any anchor points, apart from the experience and knowledge of language teachers, different EPs of the same canonical pronunciation are hard to locate concerning the correct pronunciation (homogeneous) in the conceptual pronunciation space, let alone to describe them in a human-understandable way, which also makes it almost impossible for human annotators to create target labels. Heterogenous initialization provides the ground to describe the error patterns, which facilitates human understanding as well as acoustic modeling.

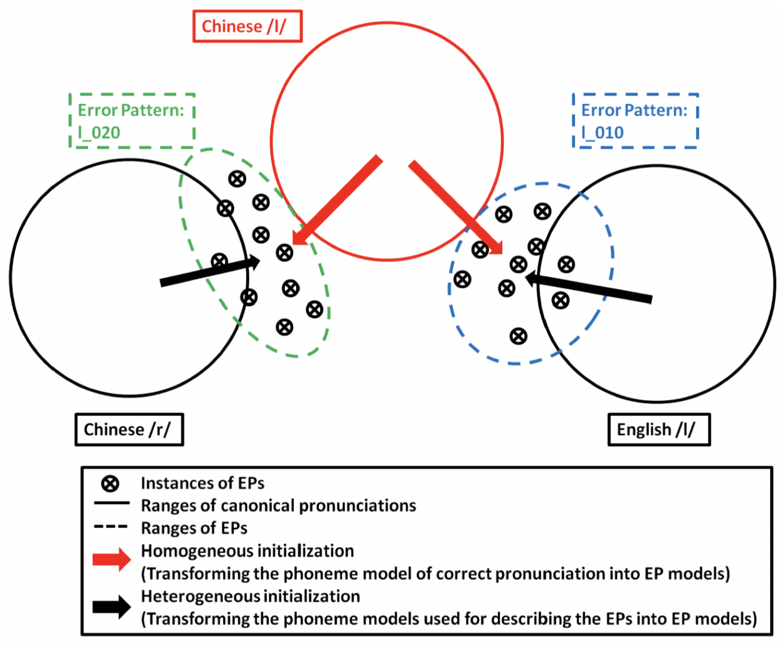
Chinese /l/
Error Pattern: l_020
Error Pattern: l_010
Chinese /r/
English /l/
Instances of EPs
Ranges of canonical pronunciations
Ranges of EPs
Homogeneous initialization
(Transforming the phoneme model of correct pronunciation into EP models)
Heterogeneous initialization
(Transforming the phoneme models used for describing the EPs into EP models)

Figure 3: Homogeneous and heterogeneous initialization for acoustic modeling of EPs in the conceptual pronunciation space, adopted from Wang and Lee (2015)

## 2.3 *From Perceptrons to Multilayer Perceptrons*

A **Perceptron** is a single-neuron model that acts as a **binary linear classifier**. It is inspired by the biological neuron and can be seen as an abstraction of it. The perceptron receives a real-valued feature vector as input, it also has a weight vector associated with the input vector, which is to be trained. A weighted sum is calculated and passed into an activation function before the output layer. There are various kinds of activation functions, they help normalize the weighted sum to a range between 0 and 1, or, in some cases, -1 and 1. Like the biological neuron needs an action potential above the threshold of -55 mA to "fire", the perceptron neuron uses the activation function and decides whether to pass on the output or not (acting as a gate), or, in the case of a binary classifier, it decides if the input belongs to a class or not. In the case of the example below, a step function is used where 0 is the threshold, everything above 0 is mapped to 1 and 0 otherwise.

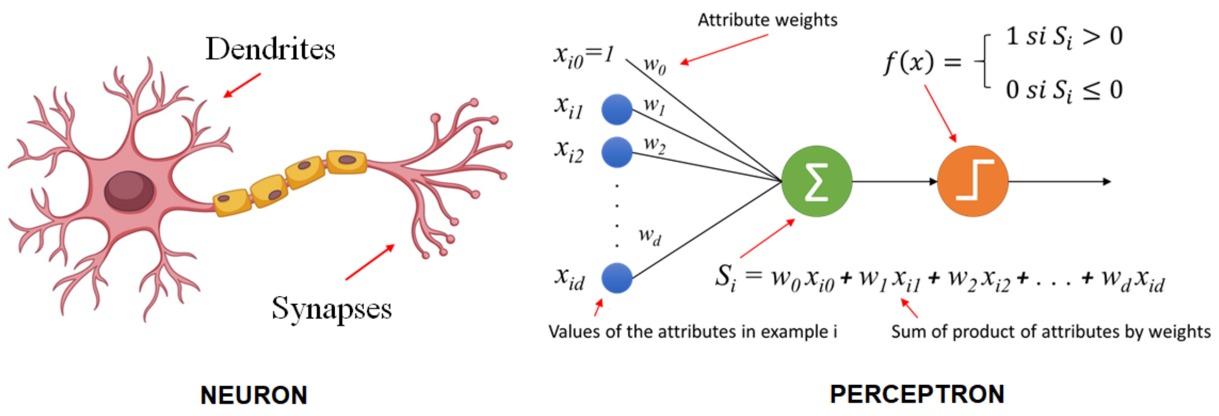
Dendrites
Synapses
NEURON
Attribute weights
$x_{i0}=1$
$w_0$
$x_{i1}$
$w_1$
$x_{i2}$
$w_2$
$w_d$
$x_{id}$
$f(x) = \begin{cases} 1 \; si \; S_i > 0 \\ 0 \; si \; S_i \leq 0 \end{cases}$
$S_i = w_0 x_{i0} + w_1 x_{i1} + w_2 x_{i2} + \ldots + w_d x_{id}$
Values of the attributes in example i
Sum of product of attributes by weights
PERCEPTRON

Figure 4: biological neuron vs. perceptron, adopted from Blog Inteligencia Futura (2019)

As suggested by the name, **multilayer perceptrons** (MLPs) have more than one (usually fully connected) layer. The most simple schematic MLP structure illustrated in Figure 5 has one input layer, a hidden layer, and an output layer. There can be more than one hidden layer, but usually one is enough. MLPs are a class of feedforward artificial neural network (ANN), with "**feedforward**" meaning single-way, acyclic movement, just like biological neurons.

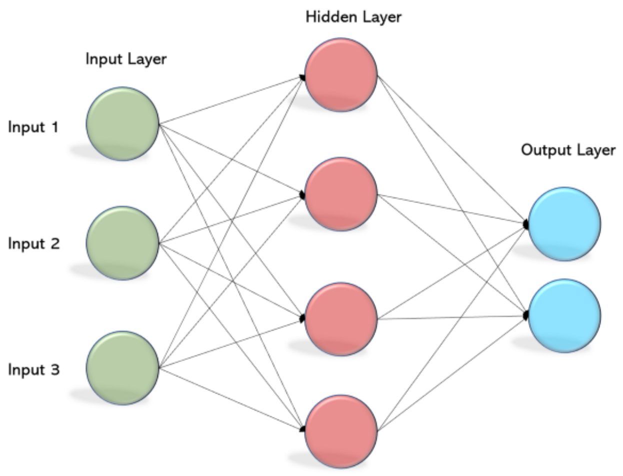

Figure 5: A schematic multilayer perceptron model, adopted from Mohanty (2019)

MLPs use **backpropagation** (BP) as training method, which compares the results to the desired value and adjust the weights that contributed to the error most in each layer using the chain rule. The adjusted weights are used in the next training cycle. The training process terminates when the loss function is minimized.

## 2.4 *Universal Phoneme Posteriorgram*

**Universal Phoneme Posteriorgram** (UPP) is used as the fundamental frame-level feature of supervised EP discovery in Wang and Lee (2015). The motivation behind this has to do with the nature of EPs in language learning. Usually, EPs are caused by articulator mechanisms or acoustic phenomena present in the target language but missing from learners' native languages as suggested by Wang and Lee (2015). However, the discrepancies between mispronounced instances and canonical pronunciations can be small but still obvious to the native ear. Therefore if one simply compares the pronunciations from learners of mixed languages in the same acoustic space, for example, the acoustic space spanned by MFCCs, the pronunciations are usually too close to be distinguished by clustering algorithms. What UPP extraction does is to use an MLP as a posterior probability estimator to span the pronunciation instances by phoneme posteriors. Each frame of MFCC feature is transformed into a vector of posterior probabilities of all predefined phonemes in the training corpora of mixed languages.

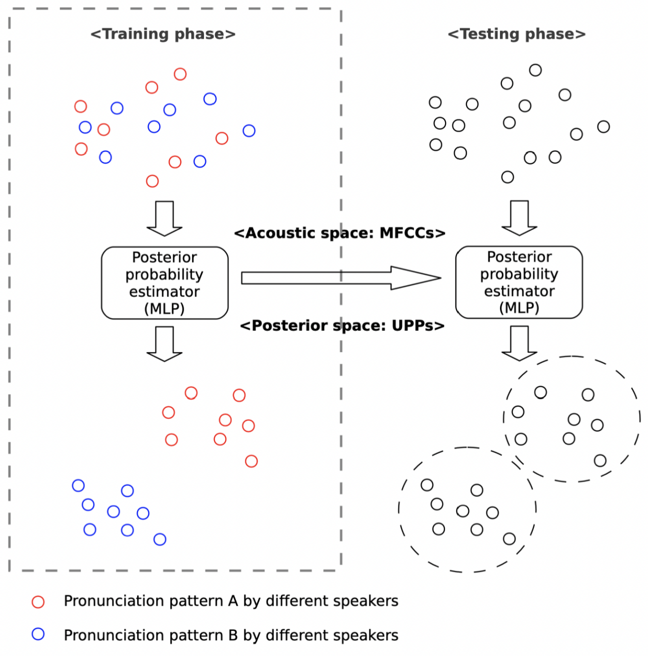
<Training phase>
<Testing phase>
<Acoustic space: MFCCs>
Posterior probability estimator (MLP)
Posterior probability estimator (MLP)
<Posterior space: UPPs>
Pronunciation pattern A by different speakers
Pronunciation pattern B by different speakers

Figure 6: The concept of **UPP extraction**: mapping from acoustic space (represented by MFCCs) to posterior space (represented by UPPs) using a posterior probability estimator (MLP) so as to better distinguish the pronunciation patterns, adopted from Wang and Lee (2015)

## 2.5 *Hidden Markov Model and Viterbi Decoding*

### 2.5.1 *Markov Chains and HMM*

**Markov chains** model the **probabilities of sequences** of random variables, which are usually termed *states*. The power of the Markov chain lies in the Markov assumption that when predicting the next state, only the current state matters[9]. Figure 7 shows a simple Markov chain that models the weather. Each circle represents a state, the arrows represent state transition, with the digits being the transition probability. Let's say today is cold, then we focus on the COLD state, there are 3 arrows pointing out from this state, so tomorrow (next state) is either COLD again with probability of 0.8, HOT or WARM, each with probability of 0.1. The probabilities of all possible transitions sum to 1.

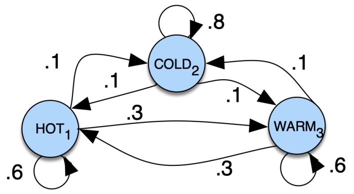

Figure 7: A Markov chain for weather, showing states and transitions, adopted from Jurafsky and Martin (2020)

[9] When predicting the future, the past doesn't matter, only the present does.

**Hidden Markov Models** (HMMs) are Markov chains with ***hidden states***. Markov chain is useful when we need to compute the probability for a sequence of observable events, but when what we are interested in cannot be observed directly and we know that it is related to what can be observed, then we use HMM. For example, Jason, as a normal boy, enjoys ice-cream. Although he has ice-cream every day, the amount of ice-creams he has each day differs. Since he is normal, he is more likely to have more ice-creams on hot days and less on cold days. Figure 8 shows an HMM that models Jason's ice-cream eating behavior and the weather. The octagon shows the distribution of the hidden states[10]. The squares are the ***emission probabilities*** or ***observation likelihoods***[11]. The rest is the same as the Markov chain explained above.

[10] This distribution can be translated as follows: On any day, the probability that it is a hot day is 0.8, the probability that it is a cold day is 0.2.

[11] Square $B_1$ can be translated as: On a hot day, it is 20% chance that Jason eats one ice-cream, 40% that he eats 2 and 40% that he eats 3.

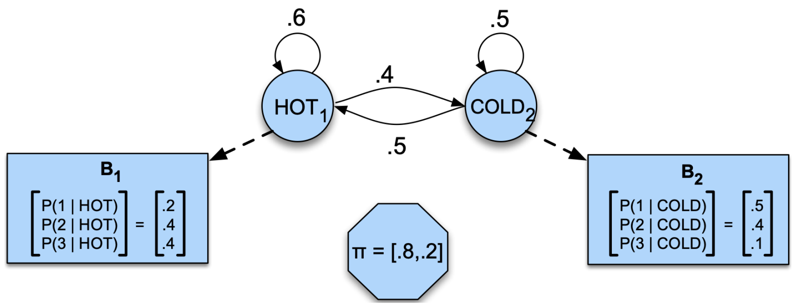
.6
.5
.4
HOT$_1$
COLD$_2$
.5
$B_1$
$\begin{bmatrix} P(1 \mid HOT) \\ P(2 \mid HOT) \\ P(3 \mid HOT) \end{bmatrix} = \begin{bmatrix} .2 \\ .4 \\ .4 \end{bmatrix}$
$B_2$
$\begin{bmatrix} P(1 \mid COLD) \\ P(2 \mid COLD) \\ P(3 \mid COLD) \end{bmatrix} = \begin{bmatrix} .5 \\ .4 \\ .1 \end{bmatrix}$
$\pi = [.8,.2]$

Figure 8: A simplistic HMM for relating numbers of ice-creams consumed by Jason each day (observations) to the weather (*Hot(H)* or *Cold(C)*, hidden states), adopted from Jurafsky and Martin (2020)

Hidden Markov Model (HMM) in Speech Recognition is used to model units of speech, like words or phones.

### 2.5.2 *Viterbi Decoding*

One of the problems that HMM can solve is the **decoding problem**:

***Given an observation sequence and an HMM, find the best hidden state sequence.***

Using the ice-cream example above, it means that if for the last 3 days the number of ice-creams Jason had is the observed sequence 3 1 3, what are the weather conditions of these three days most likely like?

**Viterbi algorithm** solves this decoding problem by finding (and remembering) the best (most likely) path at each step. Figure 9 shows the observations in squares and hidden states in circles. All possible states are drawn, but only the current most likely state is chosen to form the optimal path. For instance, since the initial weather distribution for (HOT, COLD) is (0.8, 0.2), and on the first day, Jason had 3 ice-creams, the probability of the hidden state being HOT is the product of 0.8 and the observation likelihood[12]. Similarly, we can calculate the probability of the hidden state being COLD. Since HOT is 0.32 and COLD is 0.02, we pick HOT. For the next steps, we don't need the initial weather distribution $\pi$ anymore, we should only consider the current hidden state and the transition to the next possible hidden states. Using Viterbi decoding, we obtain the thick arrows as the optimal path.

[12] the probability of Jason eating 3 ice-creams given it is a hot day

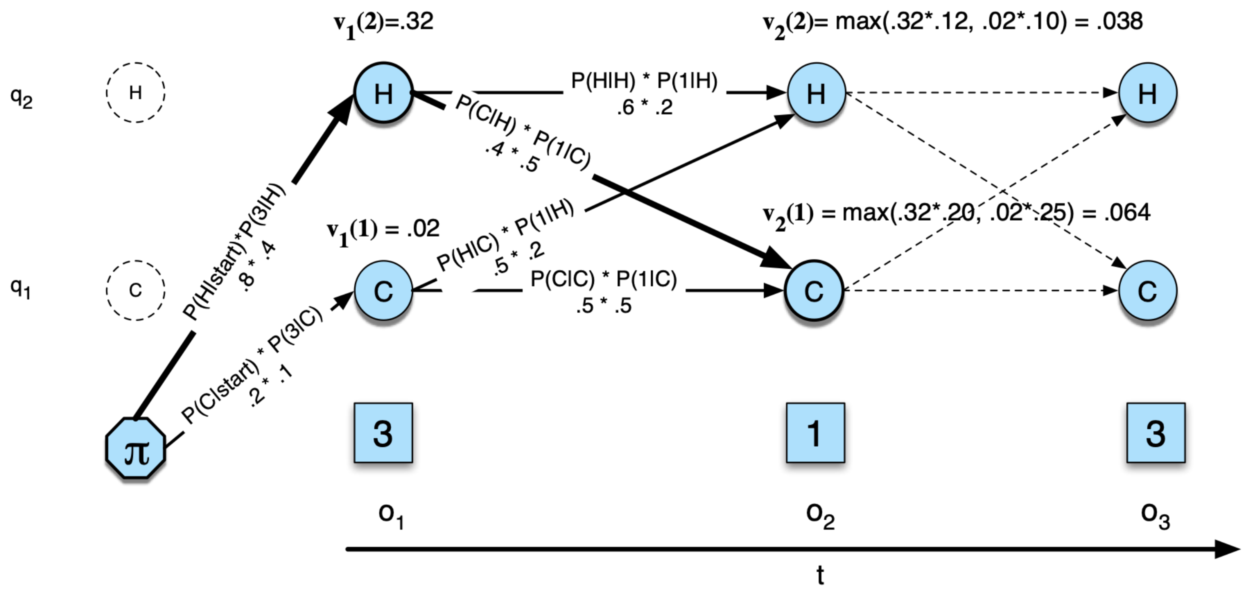

Figure 9: Viterbi trellis for computing the best path through the hidden state space for the ice-cream eating events 3-1-3. Hidden states are in circles, observations in squares. This figure is adopted from Jurafsky and Martin (2020)

## 3. *Methods*

### 3.1 *Clustering*

Before starting with Clustering, we should understand first what is a cluster. [13] Simply speaking, **Cluster** is the collection of data. The data objects in one cluster are similar to one another within the same group (class or category), and are different from the objects in the other clusters. The closer the objects in a cluster, the more likely they belong to the same cluster.

[13] This part is based on Seema Singh (2018) "An Introduction To Clustering". Here is the Web link `https://medium.datadriveninvestor.com/an-introduction-to-clustering-61f6930e3e0b`

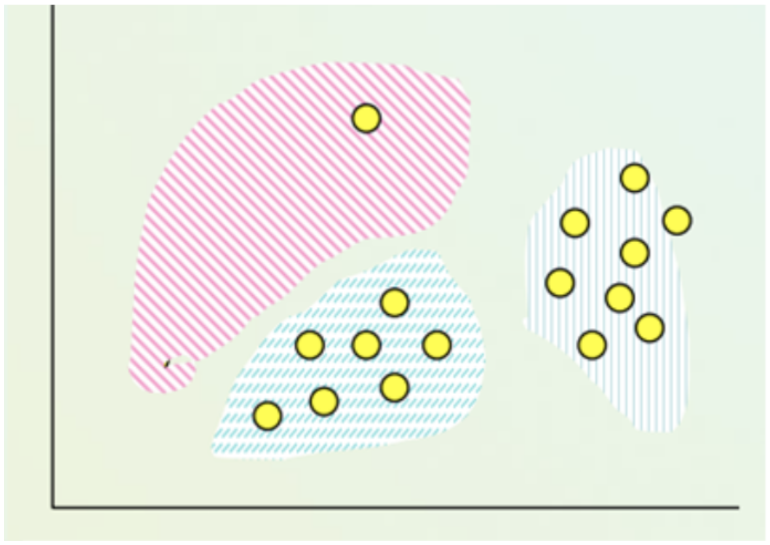

Figure 10: clustering, adopted from `https://medium.datadriveninvestor.com/an-introduction-to-clustering-61f6930e3e0b`

Based on the definition of the Cluster, as an unsupervised learning technique in which there are predefined classes and prior information, Clustering means how the data should be grouped or labeled into separate classes. It could also be considered as Exploratory Data Analysis (EDA) [14] process which help us to discover hidden patterns of interest or structure in data. Clustering can work as a standalone tool to get insights about the data distribution or as a preprocessing step in other algorithms.

[14] Exploratory Data Analysis refers to the critical process of performing initial investigations on data to discover patterns to spot anomalies to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

In Wang and Lee (2015), they uses **Hierarchical Agglomerative Clustering** (HAC) to define segment-level features. According to Chhabra and Mohapatra (2020), Hierarchical Agglomerative Clustering (HAC) refers to a class of greedy unsupervised learning algorithms that seek to build a hierarchy between data points while clustering them in a bottom-up fashion. The HAC algorithm used in Wang and Lee (2015) automatically arranges the frames in a speech segment into a tree-structured hierarchy based on the merging order of sub-segments, in which similar adjacent frames are clustered together in lower layers, while relatively dissimilar adjacent clusters are merged in higher layers.

## 3.2 *Metrics*

There are many different metrics for evaluating clustering algorithms. Metrics are measures of quantitative assessment commonly used for comparing and tracking performance or production. Metrics can be used in a variety of scenarios. Wang and Lee (2015) define the pair-wise true acceptance (TA′), true rejection (TR′), false acceptance (FA′), and false rejection (FR′) for clustering tasks based on all instance pairs. Let us explain these by one.

The True Accept rate (TA′) is a statistic used to measure biometric performance when performing the verification task. It is the percentage of times a system (correctly) verifies a true claim of identity. While the True Reject rate (TR′) is a statistic used to measure biometric performance when performing the verification task. It refers to the percentage of times a system (correctly) rejects a false claim of identity.

False Acceptance (FA′) is the percentage of identification instances in which unauthorized persons are incorrectly accepted, and it occurs when an unauthorized subject is accepted as valid. While the False Rejection (FR′) is the percentage of identification instances in which authorized persons are incorrectly rejected. As the number of false acceptances (FA′) goes down, the number of false rejections (FR′) will go up and vice versa.

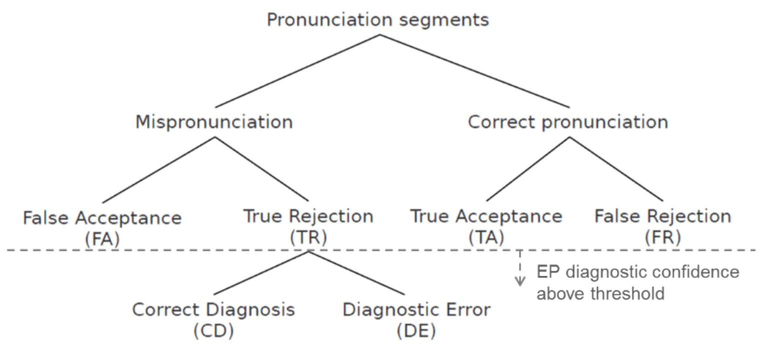

Figure 11: Hierarchical structure of the metrics used in EP detection adopted from Wang and Lee (2015)

the

## *References*

Blog Inteligencia Futura (2019). Deep neural networks, or Perceptron vs dogs and cats. [Online; accessed March 25, 2021].

Chhabra, A. and Mohapatra, P. (2020). Fair algorithms for hierarchical agglomerative clustering. *arXiv preprint arXiv:2005.03197*.

Jurafsky, D. and Martin, J. H. (2020). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 3rd edition. [Online; accessed March 29, 2021].

Mohanty, A. (2019). Multi layer Perceptron (MLP) Models on Real World Banking Data. [Online; accessed March 25, 2021].

Wang, Y.-B. and Lee, L.-s. (2015). Supervised detection and unsupervised discovery of pronunciation error patterns for computer-assisted language learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):564–579.