Dominant Speaker Detection in Multipoint Video Communication Using Markov Chain with Non-Linear Weights and Dynamic Transition Window

Vishnu Monn Baskaran[a,b,*], Yoong Choon Chang[c], Jonathan Loo[d], KokSheik Wong[b], MingTao Gan[e]

[a]*Electrical and Computer Systems Engineering, School of Engineering, Monash University Malaysia, Jalan Lagoon Selatan, Bandar Sunway, 47500 Subang Jaya, Selangor, Malaysia*
[b]*School of Information Technology, Monash University Malaysia, Jalan Lagoon Selatan, Bandar Sunway, 47500 Subang Jaya, Selangor, Malaysia*
[c]*Lee Kong Chian Faculty of Engineering & Science, University Tunku Abdul Rahman, Bandar Sungai Long, Cheras, 43000 Kajang, Selangor, Malaysia*
[d]*School of Computing & Engineering, University of West London, St Mary's Rd, London W5 5RF, U.K*
[e]*Faculty of Engineering, Multimedia University, Persiaran Multimedia, 63100 Cyberjaya, Selangor, Malaysia*

Abstract

This paper proposes an enhanced discrete-time Markov chain algorithm in predicting dominant speaker(s) for multipoint video communication system in the presence of transient speech. The proposed algorithm exploits statistical properties of the past speech patterns to accurately predict the dominant speaker for the next time state. Non-linear weights-based coefficients are employed in the enhanced Markov chain for both the initial state vector and transition probability matrix. These weights significantly improve the time taken to predict a new dominant speaker during a conference session. In addition, a mechanism to dynamically modify the size of the transition probability matrix window/container is introduced to improve the adaptability of the Markov chain towards the variability of speech characteristics. Simulation results indicate that for an 11 conference participants test scenario, the enhanced Markov chain prediction algorithm registered an 85% accuracy in predicting a dominant speaker when compared to an ideal case where there is no transient speech. Misclassification of dominant speakers due to transient speech was also reduced by 87%.

Keywords: Markov chain, dominant speaker detection, multipoint video communication

*Corresponding author
Email addresses: vishnu.monn@monash.edu (Vishnu Monn Baskaran), ycchang@utar.edu.my (Yoong Choon Chang), jonathan.loo@uwl.ac.uk (Jonathan Loo), wong.koksheik@monash.edu (KokSheik Wong), mtgan@mmu.edu.my (MingTao Gan)

1. Introduction

Multipoint video communication (MVC) captures and transmits twoway audio signals and motion images in real-time across vast distances and different time zones. It serves as a mean to bring us closer together albeit being physically apart, hence increasing the efficiency in human communication. This motivates MVC to continuously evolve through improvements in real-time video delivery codecs, high speed intercontinental networks and advanced computing architectures. Today, MVC is matured enough be deployed in a wireline network environment where a high quality of service (QoS) is sustainable for an immersive user experience. On top of that, MVC is rapidly trending towards mobile wireless environment, largely contributed by the increasing popularity of the mobile office concept [15, 8]. For instance, MVC plays a pivotal role in enabling an immersive platform for boardroom meetings on the move.

To this end, a mobile MVC relies on a wireless internet infrastructure. This reliance, however, is by no means challenge-free as operating in a mobile environment has its constrains. One such constrain applies to limited network bandwidth especially when a large number of mobile users are simultaneously connected. To guarantee fair bandwidth utilization, mobile operators may be compelled to implement congestion control techniques. However, this may lead to a throttled [4] or capped bandwidth [5, 36, 23, 18]. In turn, this gives lower bit rates to a subscriber who could have otherwise experienced a high QoS equivalent to a wireline network. The risk of a throttled/capped bandwidth during a lengthy MVC session requires the need to regulate its bandwidth consumption. With proper regulation, a mobile MVC system is able to extend its usage duration with high QoS that is equivalent to a wireline network.

One method here is to implement an unequal bitrate distribution of video streams to each conference participant, whereby the loudest speaker/s is/are allocated with a higher portion of a regulated bandwidth (i.e., speaker selection) [9]. Typically in a MVC session, the viewers attention would be focused to the client who is speaking in a talkspurt, which is referred to as the dominant speaker. Therefore, a speaker selection process allocates a higher percentage of regulated bandwidth to the dominant speaker. The impact of this allocation works in two ways. First, in continuous presence based MVC systems, the dominant speaker is allocated with significant portions of a regulated bandwidth, which translates into higher coding rates and improved audio and visual clarity. Second, in voice activated switching MVC systems, on top of higher coding rates, the dominant speaker is also allocated with larger portions of the display resolution [10].

However, MVC systems, especially in the presence of a large number of connected participants exhibits variability in speech characteristics. A significant portion of this variability is due to transient speech or noise patterns. A transient speech is defined as a burst of speech lasting for a very short duration, which may be mistaken for a dominant speaker (henceforth referred to as false dominant speaker). Critically, the false classification of a dominant speaker results in incorrect unequal bitrate distribution such that the genuine dominant

speaker is allocated with a smaller rate density and subsequently reduced audio and visual clarity. The impact is more significant in voice activated switching based video communication systems where false detection of a dominant speaker would result in false switching between speakers on a display system. It is undeniable that for a regulated network bandwidth, the speaker selection process is crucial in a MVC system. Nevertheless, the variability of speech characteristics necessitates a method to minimize the impact of transient speech in misclassifying a dominant speaker and its consequent incorrect bitrate distribution.

Therefore, in this paper, an enhanced discrete-time Markov chain algorithm is proposed to predict dominant speaker(s) in a MVC system. First, this algorithm analyzes the loudness (or amplitude) of speaker(s) at each client endpoint in a conference session to determine the loudest speaker at a specific point of time. The loudest speaker at current and previous points of time are then evaluated to predict the dominant speaker. The aim here is to maximize prediction accuracy of a dominant speaker and minimize the impact of transient speech on false dominant speaker classification. The contributions of this paper are summarized as follows:

1. A discrete Markov chain algorithm is applied to analyze statistical properties of past speech patterns of the loudest speakers at the present time to accurately predict the dominant speaker for the next time state;
2. Non-linear weights-based coefficients are assigned for both the initial state vector and transition probability matrix of a Markov chain, which significantly improve the responsiveness of the algorithm towards changes in dominant speakers, and;
3. An original mechanism that dynamically modifies the size of a transition probability matrix container is implemented, whereby a confidence interval parameter is utilized to determine an ideal container size during a conference session. This method improves the adaptability of the Markov chain algorithm towards the variability in speech characteristics. The proposed enhanced discrete-time Markov chain algorithm is able to reliably predict a dominant speaker and significantly reduce misclassification of dominant speakers in the presence of transient speech.

The rest of the paper is organized as follows: Section 2 reviews the related work. Section 3 formulates the problem description in dominant speaker identification with Section 4 describing the enhanced Markov chain algorithm with weights-based coefficients. Section 5 models an adaptive transition matrix container. Section 6 describes the simulation environment. Section 7 analyzes the performance of the enhanced Markov Chain algorithm and Section 8 concludes this paper.

2. Related Work

A multitude of research have been undertaken to identify a dominant speaker as a basic mechanism of social interaction, who would then lead the group con-

versation and becomes the focus of the conversation [17]. In fact, dominant speaker detection represents a subset of speaker diarization [1]. Table 1 summarizes related work to this research. Earlier work in this field was based on a psychological perspective whereby the level of social psychological influence of each participant in a meeting were calculated and ranked, with the most dominant person recognized as the one with the highest influence level [20, 27, 28]. Aside from a psychological perspective, audio and video features plus nonverbal activity cues were also applied in recognizing the most dominant speaker in a face-to-face (or physical) meeting [14, 13, 16].

The aforementioned work on dominant speaker recognition were focused for group/physical meetings within a single locality. However, when multiple localities are factored in for a virtual meeting through the use of a MVC system, the purpose in identifying a dominant speaker differs from the aforementioned work. Specifically, in voice activated switching based MVC systems, a common aim is evident in comparison to the aforementioned work where the identified dominant speaker is allocated with a larger portion of screen resolution representing the focus of the conversation. However, for both continuous presence and voice activated switching based MVC systems, identifying a dominant speaker allows the conference engine to optimize data traffic for conference participants in the presence of a regulated network bandwidth.

In addition, identifying a dominant speaker requires periodical analysis of conversational patterns from different clients and the ensuing unequal rate control. The rate control typically applies a form of foveating such that the visual clarity of a dominant speaker will appear sharper, relative to the non-dominant speakers [29]. Research into unequal rate control for a dominant speaker applied dynamic bit allocation and dynamic region of interest transcoding [31, 32, 19, 11]. These methods function on the presumption that a MVC session would typically have one or two active (or dominant) speakers at one time. The dominant speakers are identified purely based on analyzing the level of motion activity of a coded participant stream. The concern here is that these methods rely solely on motion activities of a speakers transmitted video frame, which may be inadequate given that high motion activity does not necessarily indicate that a participant is a dominant speaker.

Hence, research was shifted towards using audio information to identify a dominant speaker during a MVC session. A typical approach was to compute the average amplitude of samples from each input audio channel at a target time interval. The speaker with the largest average amplitude is classified as the dominant speaker [34]. However, this method is susceptible towards the impact of environmental noise in an audio channel. To improve resistance towards noise, enhancements were applied by incorporating an automatic gain controller, which includes a weighted computation of current and past average amplitude samples from each input audio channel [22, 3]. Alternatively, analytics were also applied in each audio channel to detect speech activities [24, 35].

Although speech detection techniques isolate noise from speech content, the usage of instantaneous instead of long term-properties of audio information risks misclassifying a dominant speaker. For instance, if a non-dominant speaker

barges into an ongoing conference conversation at a particular time interval to the extent that the non-dominant speakers speech signal constitutes the highest amplitude among other speakers, this speaker would then be classified as a dominant speaker. In actuality though, the burst is only temporary and does not necessarily warrant a dominant speaker switch. The usage of long term audio information properties have been used for voice activity detection [30, 26, 25, 6, 7], but these properties were not originally considered in identifying a dominant speaker.

To this end, Volfin & Cohen proposed a dominant speaker identification method using long term audio information by evaluating speech activity of different lengths [33]. Specifically, speech activity scores of each conference speaker are evaluated for the immediate, medium and long time intervals. These scores are used as parameters to a likelihood function in a loglikelihood ratio computation, which are then compared with a set of pre-defined thresholds to identify a dominant speaker. A second score evaluation procedure using hidden Markov model in the likelihood function was also considered to detect the presence of speech. The tolerable transition delay from one dominant speaker to a new dominant speaker is set at 1 second(s). Hence, the observed time frame of an audio information does not exceed the 1s boundary. Obviously, if a speaker accidentally interrupts the conversation (e.g., coughing, laughter) of a dominant speaker to the extent that the speech burst lasts beyond 1s, this algorithm would classify the source of the transient speech as a dominant speaker. Another factor often overlooked would be the conversation exchange between conference participants. As highlighted above, the general presumption is that one or two participants are active at specific times during a MVC session. Based on this presumption, a dominant speaker identification algorithm needs to be able to analyze conversational patterns between speakers, enabling it to classify a barge in by another speaker as either a genuine conversational response to the current dominant speaker or as transient speech.

While the aforementioned literature lay a solid groundwork in dominant speaker identification, the fact of the matter is that the variability in speech characteristics means that it would be infeasible to fix the size of a transient speech length. It is vital that a dominant speaker identification algorithm is able to reliably classify a dominant speaker under varying transient speech lengths, and this issue remains unresolved even in the context of speaker diarization [1]. Therefore, this paper utilizes a discrete-time Markov chain algorithm to evaluate current and past speech activities of conference participants in identifying a dominant speaker. To emphasize, in the aforementioned literature, Markov chains were used for speech detection. In this paper, the Markov chain is used to determine the transition of dominant speakers over a timeframe larger than 1s. Crucially, the proposed enhanced Markov chain algorithm is able to reliably classify a dominant speaker under varying transient speech lengths. The following sections formulate the problem statement, Markov chain implementation and the proposed enhancements.

Table 1: Summary of related work

Author/citation	Environment (Physical Meeting, Virtual Meeting)	Objective	Method
Jie and Peng, 2010 [17]	Physical Meeting	Social interaction analysis	Psychological analysis
Mast, 2002 [20]	Physical Meeting	Psychological influence	Inference through speaking duration
Rienks and Heylen, 2005 [27]	Physical Meeting	Psychological influence	Verbal and nonverbal activity cues detection
Rienks et al., 2006 [28]	Physical Meeting	Psychological influence	Verbal and nonverbal activity cues detection
Hung et al., 2007 [14]	Physical Meeting	Psychological influence	Audio and video features, and nonverbal activity cues
Hung et al., 2008 [13]	Physical Meeting	Psychological influence	Audio and video features, and nonverbal activity cues
Jayagopi et al., 2009 [16]	Physical Meeting	Psychological influence	Audio and video features, and nonverbal activity cues
Sun et al., (1997) [31]	Virtual Meeting	Speaker selection	Coded domain video stitching
Sun et al., (1998) [32]	Virtual Meeting	Speaker selection	Dynamic bit allocation
Lin et al., (2003) [19]	Virtual Meeting	Speaker selection	Dynamic region of interest transcoding
Fung et al., (2004) [11]	Virtual Meeting	Speaker selection	Frame skipping transcoder
Xing et al., (2005) [34]	Virtual Meeting	Speaker selection	Adaptive audio mixing
Nagata et al., (2006) [22]	Virtual Meeting	Speaker selection	Auto gain controller
Baskaran et al., (2010) [3]	Virtual Meeting	Speaker selection	Auto gain controller
Ramrez et al., (2007) [24]	Virtual Meeting	Speaker selection	Speech recognition
Xu et al., (2006) [35]	Virtual Meeting	Speaker selection	Silence suppression
Sohn et al., (1999) [30]	Virtual Meeting	Voice activity detection	Statistical model technique
Ramrez et al., (2004) [26]	Virtual Meeting	Voice activity detection	Long term speech information
Ramrez et al., (2005) [25]	Virtual Meeting	Voice activity detection	Multiple observation likelihood ratio test
Dove et al. (2015) [6]	Virtual Meeting	Voice activity detection	Diffusion maps
Dove et al. (2016) [7]	Virtual Meeting	Voice activity detection	Kernel method
Volfin et al. (2013) [33]	Virtual Meeting	Dominant speaker identification	Loglikelihood method

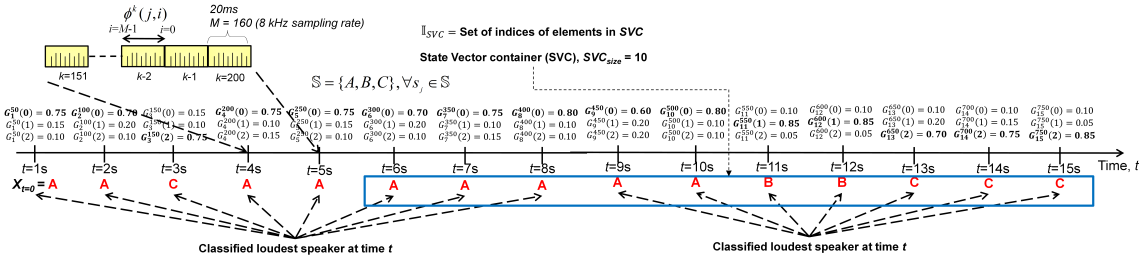


Figure 1: Sample distribution of loudest speaker for 3 clients in a conference session at each t, up to $t = 15$ seconds.

3. Problem Formulation

This section formulates a method to identify a loudest speaker at a time interval and argues the susceptibility of this method towards misclassification errors due to varying transient speech lengths. Table 2 summarizes the list of notations that are applied in the current following sections of this paper. Let $\mathbb{S}$ be a finite set of states representing the various clients in a MVC session. At any time t, the dominant speaker is a random variable X_t, where t is a series of discrete time points in a parametric time space $\mathbb{T}$, such that $t_0 < t_1 < \cdots < t_i < \cdots < t_n \in \mathbb{T}$. Fig. 1 illustrates a sample time chart of 3 clients in a conference session with $\mathbb{S} = \{A, B, C\}$.

A time duration of 15 seconds is illustrated, separated into one-second time intervals. At each t, the loudest speaker X_t is identified such that this speaker will be allocated with a majority of available network bandwidth. The identification of loudest speaker at time t can be achieved via several methods, as discussed in the aforementioned section. Here, a gain controller algorithm is applied where a gain function $G_t^k(j)$ is defined such that for $\forall s_j \in \mathbb{S}$,

$$G_t^k(j) = \frac{1}{L}\left(\frac{A_t^k(j)}{\sum_{z\in\mathbb{S}\setminus\{s_j\}} A_t^k(z)} + \sum_{l=1}^{L-1} G_t^{k-l}(j)\right) \tag{1}$$

$$and\ A_t^k(j) = \frac{1}{M}\left(\sum_{i=0}^{M-1} \phi^k(j,i)\right) \tag{2}$$

$G_t^k(j)$ is the normalized gain of s_j for the k-th audio packet, where $\sum_{i\in\mathbb{S}} G_t^k(j) = 1$. $A_t^k(j)$ is the average amplitude (i.e., channel power) of the audio samples of s_j in k-th audio packet. $\phi^k(j,i)$ is the i-th sample in the k-th packet of s_j and M represents the number of samples in an audio packet. In Fig. 1, $M = 160$ for a given sampling rate of 8 kHz (i.e., 8000 samples/sec). The audio stream transmission is typically based on a real-time transport protocol (RTP) with each RTP packet having a length of 20ms. As such, based on a 1s sampling interval, $k = 50t$. Equation (1) aligns the gain based on the greatest channel

Table 2: Summary of notations applied in this paper

Notations	Description
$\mathbb{S}$	Set of clients in a multipoint video communication session.
s_j	j-th client video, $s_j \in \mathbb{S}$.
$\mathbb{T}$	Parametric time space, $t \in \mathbb{S}$.
$G_t^k(j)$	Normalized gain of s_j for the k-th audio packet.
$A_t^k(j)$	Average amplitude of the audio samples of s_j in the k-th audio packet.
X_t	Loudest speaker at time t.
$\boldsymbol{\pi}(t)$	State probability vector at time t.
$P_{ij}(t)$	One-step transition probability at time t.
$\mathbf{P}(t)$	One-step transition probability matrix at time t.
$\mathbb{I}_{SVC}$	Set of indices of elements in a state vector container.
$\mathbb{I}_{TMC}$	Set of indices of elements in a transition matrix container.
$\mathbb{I}_{OSVC}$	Set of indices of elements in an observed state vector container.
$W_{s_j}(i)$	Weighting function of each s_j at the i-th index in a state vector container, where $i \in \mathbb{I}_{SVC}$
D_t	Dominant speaker at time t.
$c(t)$	Size of transition matrix container at time t.
$u(t)$	Number of older (or earlier) elements which are either factored into or removed from the TMC at time t.
${}^{obs}\pi(t)$	Observed (obs) state vector at time t.
$\mathbb{Q}(t)$	Set of transition matrix container sizes at time t, where $\mathbb{Q} = \{c(t) - u(t), c(t), c(t) + u(t)\}$ and $q_l \in \mathbb{Q}(t)$.
${}^{q_l}\varphi(t)$	Average of the predicted state probability vectors.
d_l	Distance between an observed state vector, ${}^{obs}\pi(t+v)$ and the average state probability vectors, ${}^{q_l}\varphi(t)$.

power of an endpoint client. This equation also factors in L packets in computing the loudest speaker at t by using a moving average window, where L represents the size of this window. Since sampling interval is 1s, $L = 50$. The computed $G_t^k(j)$ would typically be multiplied with s_j during an audio mixing process. However, $G_t^k(j)$ is used here to determine the loudest speaker X_t. Here, the endpoint client (i.e., s_j) with the highest $G_t^k(j)$ at time t is determined as the loudest speaker. As such,

$$X_t = s_{j^*(t)} \tag{3}$$

where

$$j^*(t) = \arg\max_j(G_t^k(j)) \tag{4}$$

Using (1), Fig. 1 illustrates an arbitrary set of $G_t^k(j)$ at t. The primary limitation of the aforementioned method in identifying a loudest speaker is based on the lack of reference to the current and past speech activities beyond a 1s duration period. In Fig. 1, the transition of the loudest speaker from client A to C at $t = 3s$ and back to A at $t = 4s$ would suggest that at $t = 3s$, the identification of client C as the loudest speaker is actually a transient speech (i.e., false dominant speaker classification). Consequently, between $t = 3s$ and $t = 4s$, client C would be allocated with a significant portion of available bandwidth or screen resolution instead of the actual dominant speaker (i.e., client A). A similar pattern is also observed at $t = 11s$, in which client B is identified as the loudest speaker for 2 seconds. The smaller speech length ratio of client B to that of the overall speech duration (i.e., from $t = 1s$ to $t = 15s$) would also suggest that that at $t = 11s$, the identification of client B as the loudest speaker is actually a transient speech. To address this issue, the current and past speech activities need to be evaluated on a larger timeframe in classifying a dominant speaker.

A straightforward approach to address the impact of transient speech on false dominant speaker identification would be to apply a basic state vector container (SVC) in computing a probability that corresponds to the transitioning of voice priority from one dominant speaker to another. Fig. 1 includes a SVC in identifying a dominant speaker. At the start of each t, the loudest speaker X_t, is identified using (1). The identified X_t is then moved into the SVC. Data in this container are used to generate a state probability vector at time t, denoted as $\boldsymbol{\pi}(t)$, where $\sum_{i \in \mathbb{S}} \pi_i(t) = 1$. The size of the SVC (denoted by SVC_{size}) can be adjusted to provide for a smaller or larger sample of past loudest speakers. At each t, the earliest X_t entry is omitted from the container to allow for the latest X_t entry, representing a simple first in first out (FIFO) data structure. In computing the probability of each state in $\boldsymbol{\pi}(t)$, a constant weighting function is defined such that for $\forall s_j \in \mathbb{S}$,

$$W_{s_j}(i) = \begin{cases} 1, \text{ if } X_{t_i} = s_j, i \in \mathbb{I}_{SVC} \\ 0, \text{ otherwise} \end{cases} \tag{5}$$

where $\mathbb{I}_{SVC}$ represents the set of indices of elements in the SVC, with $i = 1$ and $i = SVC_{size}$ being the oldest element and most recent elements, respectively. This means that by using (5), a client with a high number of X_t entries into a SVC at time t will equally have a higher weight allocation. Equation (5) is then used to compute each element of $\boldsymbol{\pi}(t)$ by

$$\pi_{s_j}(t) = \sum_{i \in \mathbb{I}_{SVC}} W_{s_j}(i) \Big/ \sum_{i \in \mathbb{I}_{SVC}, s_k \in \mathbb{S}} W_{s_k}(i) \tag{6}$$

The dominant speaker, D_t is determined by identifying the client with the largest probability distribution in $\boldsymbol{\pi}(t)$, which is expressed as

$$D_t = s_{j^*(t)} \tag{7}$$

where

$$j^*(t) = \arg\max_j(\pi_{s_j}(t)) \tag{8}$$

In the sample illustration of Fig. 1, at $t = 15s$, $\pi(15) = \begin{matrix} A & B & C \\ [0.50 & 0.20 & 0.30] \end{matrix}$ and hence $D_{15} = A$. Based on this outcome and even although $X_{15} = C$, from $t = 15s$ to $t = 16s$, client A is classified as the dominant speaker. At $t = 16s$, this process is repeated where X_{16} is computed, which is then pushed into the SVC to compute $\boldsymbol{\pi}(16)$ and subsequently D_{16}.

Note that a similar case is also observed at $t = 11s$ and $t = 12s$ whereby Client A remains as the dominant speaker in spite of $X_{11} = X_{12} = B$ (i.e., $\boldsymbol{\pi}(12) = \begin{matrix} A & B & C \\ [0.70 & 0.20 & 0.10] \end{matrix}$). The two second speech duration of client B translates into a smaller probability distribution in $\boldsymbol{\pi}(12)$, which is insufficient to classify this duration as a dominant speaker.

The limitation of this method is the potentially longer time required to transit from one dominant speaker to another. In Fig. 1, if client C remains as the loudest speaker, it would take 5 seconds for this client to be eventually determined as the dominant speaker at $t = 17s$. From a subjective perspective, this delay may not have a significant impact for continuous presence based video communication systems. However, the impact could be visible for a voice activated switching video communication system, where a viewer would need to wait for 5 seconds to observe a switch to client C as the dominant speaker. A quick fix would be to reduce the size of SVC to allow for a faster response in a speaker switch. However, a smaller SVC would risk not being able to counter transient speech content, which is undesirable. The aim here is for a method that can reliably predict a dominant speaker with smaller transition delays from one dominant speaker to the next, but at the same time minimize the impact of transient speech in misclassification of a dominant speaker.

4. Weighted Markov Chain for Dominant Speaker Prediction

To minimize both the transition delay and misclassification of a dominant speaker, a Markov chain is applied here, which represents a discrete-time stochastic process [37] whereby the conditional probability distribution for X_{t+1} is defined as

$$P\left\{X_{t+1}=j|X_t=i\right\} \triangleq P_{ij}(t) \tag{9}$$

where $i,j \in \mathbb{S}$, $0 < P_{ij}(t) < 1$ and $\sum_{j=0}^{|\mathbb{S}|-1} P_{ij}(t) = 1$. $P_{ij}(t)$ represents a one-step transition probability at time t, which denotes the probability that the Markov chain, when in state i, moves next into state j one unit of time later (i.e., $t+1$). Since $|\mathbb{S}|$ is finite, a one-step transition probability matrix, $\mathbf{P}(t)$ is used to define the Markov chain where $\mathbf{P}(t) = (P_{ij}(t))\,,\ i,j \in \mathbb{S}$. The initial state vector, $\boldsymbol{\pi}(0)$ represents the probability distribution of the Markov chain when $t=0$. Suppose $\boldsymbol{\pi}(t)$ is known, then the state probability at time $t+1$ is predicted by

$$\pi_j(t+1) = \sum_{i=0}^{|\mathbb{S}|-1} \pi_i(t) P_{ij}(t) \tag{10}$$

Given the speech variability in a MVC session, it would be infeasible to predict with certainty the state of a Markov chain at a given point in the future. However, the statistical properties of the system's future can be predicted, in which these properties are computed and used here to identify a dominant speaker.

Fig. 2 illustrates a sample time chart of a Markov chain implementation for a conference session with 3 clients. At each t, the identified loudest speaker X_t is moved into a SVC and a transition state matrix container (TMC). These containers consist of the latest SVC_{size} and TMC_{size} loudest speakers respectively. Both the SVC and TMC operate on a FIFO data structure. Fig. 2 illustrate the case where $SVC_{size} = 5$ and $TMC_{size} = 13$. Data in the SVC is used to compute the state probability vector at time t (i.e., $\boldsymbol{\pi}(t)$). Data in the TMC are used to compute the state transition probability matrix, $\mathbf{P}(t)$. The product of $\boldsymbol{\pi}(t)$ and $\mathbf{P}(t)$ would result in a predicted state probability vector for the next time instance, $\boldsymbol{\pi}(t+1)$. To compute each element of $\boldsymbol{\pi}(t)$, a non-linear weighting function is defined such that for $\forall s_j \in \mathbb{S}$,

$$W_{s_j}(i) = \begin{cases} W_{\min} + (W_{\max} - W_{\min}) \times \left(\frac{1-e^{\frac{-\alpha(i-i_{\min})}{i_{\max}-i_{\min}}}}{1-e^{-\alpha}} \right), \\ \text{if } X_{t_i} = s_j \\ 0, \text{ otherwise} \end{cases} \tag{11}$$

where $i \in \mathbb{I}_{SVC}$, $\mathbb{I}_{SVC}$ having been defined after (5). W_{min} and W_{max} represent the smallest and largest weights values, respectively, while i_{min} and i_{max} represent the smallest and largest indices in $\mathbb{I}_{SVC}$, respectively. Typically, $W_{min} = 1$, $W_{max} = SVC_{size}$, $i_{min} = 1$ and $i_{max} = SVC_{size}$. α is the exponential decay

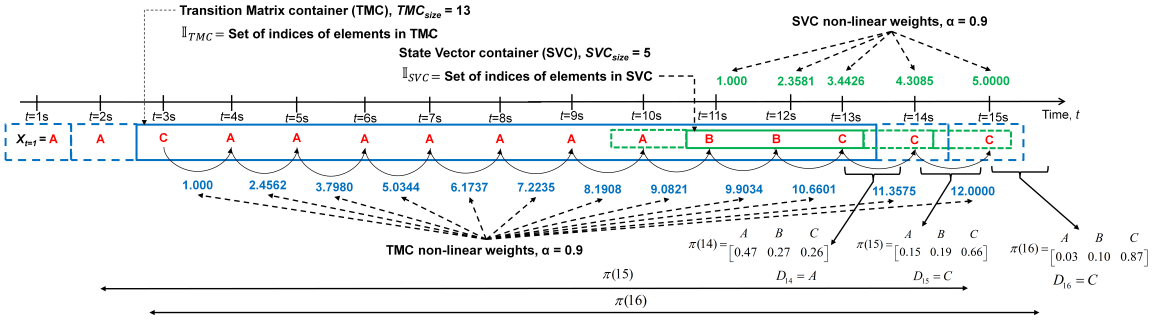


Figure 2: Sample population of SVC and TMC based on the loudest speaker at each t, up to $t = 15s$. Non-linear weights are applied for each element in the SVC and TMC, with $\alpha = 0.9$. These weights increase the responsiveness of the Markov chain towards changes in dominant speaker.

constant. Note that $\alpha \neq 0$ and $i_{\min} \neq i_{\max}$. The weights are distributed such that the most recent entry into SVC is assigned the largest weight, and these weights decrease exponentially for older entries (see Fig. 2). Equation (11) is then used to compute the state probabilities of $\boldsymbol{\pi}(t)$ as in (6).

In computing the probability of each state transition in $\mathbf{P}(t)$, a second exponential weighting function is defined as

$$W_{s_k \to s_j}(i) = \begin{cases} W_{\min} + (W_{\max} - W_{\min}) \times \left(\frac{1 - e^{\frac{-\alpha(i - i_{\min})}{i_{\max} - i_{\min}}}}{1 - e^{-\alpha}} \right), \\ \text{if } (X_{t_i}, X_{t_i - 1}) = (s_j, s_k) \\ 0, \text{ otherwise} \end{cases} \tag{12}$$

where $i \in \mathbb{I}_{TMC}$ and $\mathbb{I}_{TMC}$ represents the set of indices of elements in the TMC. i_{min} and i_{max} represent the smallest and largest indices in $\mathbb{I}_{TMC}$, respectively. Typically, $W_{min} = 1$, $W_{max} = TMC_{size} - 1$, $i_{min} = 1$ and $i_{max} = TMC_{size} - 1$. Using (12), the state transition probabilities in $\mathbf{P}(t)$ are computed as

$$P_{s_k s_j}(t) = \sum_{i \in \mathbb{I}_{TMC}} W_{s_k \to s_j}(i) \Bigg/ \sum_{i \in \mathbb{I}_{TMC}, s_l \in \mathbb{S}} W_{s_k \to s_l}(i) \tag{13}$$

where $s_k, s_j \in \mathbb{S}$. The aim here is to apply a non-linear rate of decay such that weighting values for entries in the SVC and TMC rapidly decrease from the most recent to the oldest. $W_{s_j}(i)$ and $W_{s_k \to s_j}(i)$ have been defined such that these functions should be W_{max} at $i = i_{max}$ and monotonically reduce to 1 at $i = 1$. The value of α can be modified to control the rate of change for $W(i)$. Finally, the predicted state probability vector at is obtained as

$$\boldsymbol{\pi}(t+1) = \boldsymbol{\pi}(t)\mathbf{P}(t) \tag{14}$$

Using (13) and (14), Fig. 3 illustrates a state diagram of the transition probability matrix based on the content of the TMC in Fig. 2 at $t = 15s$. In spite of a relatively smaller number of client C's inside the TMC (i.e., $X_3 = X_{13} = X_{14} = X_{15} = C$), the non-linear weight distribution applies a higher weight allocation for recent entries into the TMC, which in turn generates a larger transition probability distribution for client C (i.e., $P_{s_2 s_2} = 0.9589$). As such, $\boldsymbol{\pi}(16)$ denotes a largest probability distribution value for client C. Based on the computed elements in $\boldsymbol{\pi}(t+1)$, the dominant speaker, D_{t+1} is determined as:

$$D_{t+1} = s_{j^*(t+1)} \tag{15}$$

where

$$j^*(t+1) = \arg\max_j(\pi_{s_j}(t+1)) \tag{16}$$

In contrast with (7), (15) predicts the dominant speaker at $t+1$ based on the statistical properties of the state probability vector and the transition probability matrix at time t. Note that the non-linear weights distribution is suitable for a larger sized TMC and SVC when analyzing longer conversational patterns between different speakers. For a smaller sized SVC and TMC (see Fig. 2), a constant or linear weights distribution is instead applicable, which was described as part of a preliminary work in [2].

The Markov chain based algorithm seems to suggest a faster response time in transitioning from one dominant speaker to another. This is evident from the sample case in Fig. 2 whereby client C is classified as the dominant speaker at $t = 14s$, which represents a 2 second delay from the detection of client C as the loudest speaker at X_{13}. Comparatively, in the preceding section, a basic SVC would incur a 5 second delay in switching from client A to client C as the dominant speaker. The difference here is attributed towards the properties of a Markov chain algorithm that utilizes both a transition probability matrix and a state probability vector to compute a predicted state probability vector.

Fig. 4 illustrates the weighting function of (11) for various α, where $W_{\min} = i_{\min} = 1$ and $W_{\max} = i_{\max} = 49$. Equation (11) reduces to a linear curve when $\alpha = 0$. When $\alpha > 0$, the rate of increase for W_i is smaller than that of when $\alpha < 0$. Consequently, when $\alpha < 0$, W_i rapidly increases as i increases. This in turn contributes to a faster response time in identifying a new dominant speaker, but at the expense increasing the risk of false dominant speaker identification due to transient speech. When $\alpha > 0$, a slower rate of increase for W_i risk increasing the response time in identifying a new dominant speaker, albeit being more resilient towards the impact of transient speech patterns.

5. Dynamic Window for Adaptive Adjustment of Transition Probability Matrix

Determining an appropriate size of the TMC (i.e., TMC_{size}), is a design challenge. Setting a small TMC_{size} limits the observation to the more recent speaker transition characteristics, which make future speech patterns having

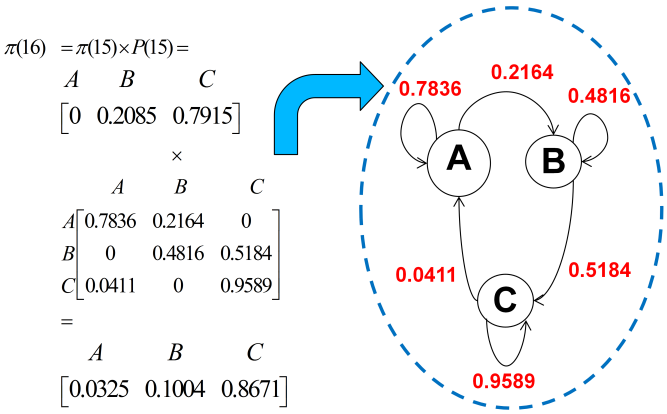


Figure 3: Transition probability state diagram at $t = 15s$ based on the sample X_t content in the SVC and TMC of Fig. 2.

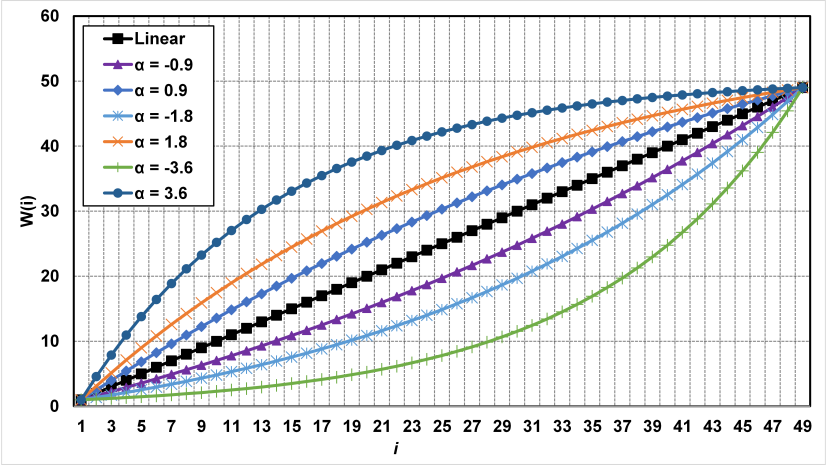


Figure 4: Linear and non-linear weight distribution at varying α values, with $W_{\min} = i_{\min} = 1$ and $W_{\max} = i_{\max} = 49$.

a higher probability of emulating. This, however, is done at the expense of increased susceptibility towards misclassification of a dominant speaker in the presence of transient speech. A larger TMC_{size} may mitigate this susceptibility, but a large TMC will include older transition characteristics which may no longer be valid for estimating future speech patterns.

In actuality, speech patterns in a MVC session are volatile for different numbers of connected conference participants at any given session. Under these circumstances, it would be infeasible to postulate a standard value for the container size. Ideally, the size of the TMC should instead be periodically adjusted

based on continuous analysis of the characteristics of a given speech pattern. To that end, this section proposes a mechanism to dynamically modify the size of a TMC container during a MVC session, by periodically analyzing the difference between an observed state vector and a set of previously predicted state vectors. The proposed method improves the adaptability of the Markov chain algorithm towards variability of speech characteristics and further reduces the time required to accurately identify the transition of one dominant speaker to another.

The concept in adaptively resizing the TMC is based on analyzing the difference between the observed state at time t and a set of previously predicted state probability vectors deduced with a reduced, maintained and expanded TMC. The TMC size yielding the predicted state probability vector that best matches the observed state is then used to compute the predicted state probability vector for $t+v$. Let $c(t)$ be the TMC_{size} at time t. At any time t, three sets of indices of elements in the TMC are generated, namely $\mathbb{I}_{TMC(c(t)-u(t))}$, $\mathbb{I}_{TMC(c(t))}$, and $\mathbb{I}_{TMC(c(t)+u(t))}$. $u(t)$ represents the number of older (or earlier) elements which are either factored into or removed from the TMC at time t. Therefore, $\mathbb{I}_{TMC(c(t)-u(t))}$ represents a shortened TMC, $\mathbb{I}_{TMC(c(t))}$ a maintained TMC and $\mathbb{I}_{TMC(c(t)+u(t))}$ an expanded TMC, all at time t. Based on these indices, the state transition probabilities are now computed as

$$ {}^{c(t)-u(t)}P_{s_k s_j}(t) = \frac{\sum\limits_{i \in \mathbb{I}_{TMC(c(t)-u(t))}} W_{s_k \to s_j}(i)}{\sum\limits_{i \in \mathbb{I}_{TMC(c(t)-u(t))}, s_l \in \mathbb{S}} W_{s_k \to s_l}(i)} \tag{17} $$

$$ {}^{c(t)}P_{s_k s_j}(t) = \frac{\sum\limits_{i \in \mathbb{I}_{TMC(c(t))}} W_{s_k \to s_j}(i)}{\sum\limits_{i \in \mathbb{I}_{TMC(c(t))}, s_l \in \mathbb{S}} W_{s_k \to s_l}(i)} \tag{18} $$

$$ {}^{c(t)+u(t)}P_{s_k s_j}(t) = \frac{\sum\limits_{i \in \mathbb{I}_{TMC(c(t)+u(t))}} W_{s_k \to s_j}(i)}{\sum\limits_{i \in \mathbb{I}_{TMC(c(t)+u(t))}, s_l \in \mathbb{S}} W_{s_k \to s_l}(i)} \tag{19} $$

To construct the observed state vector, an observed state vector container (OSVC) of size v is used to store the most recent v loudest speakers. Let $\mathbb{I}_{OSVC}$ represent the set of indices of elements in the OSVC. Note that the OSVC shares similar properties to that of the SVC. However, the size of OSVC is typically smaller, with constant weight distribution. Hence,

$$ W_{s_j}(i) = \begin{cases} 1, \text{ if } X_{t_i} = s_j, i \in \mathbb{I}_{OSVC} \\ 0, \text{ otherwise} \end{cases} \tag{20} $$

Using (20), the elements of the observed state vector are

$$ {}^{obs}\pi_{s_j}(t) = \sum_{i \in \mathbb{I}_{OSVC}} W_{s_j}(i) \Big/ \sum_{i \in \mathbb{I}_{OSVC}, s_k \in \mathbb{S}} W_{s_k}(i) \tag{21} $$

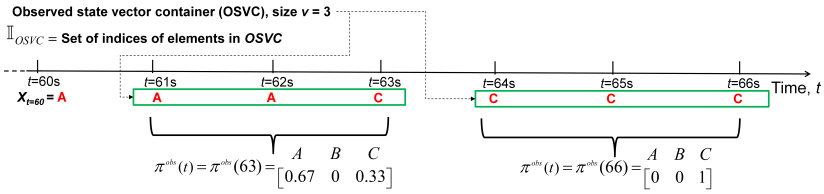


Figure 5: Sample population of OSVC based on the loudest speaker at each v interval, from $t = 60s$ to $t = 66s$. The size of OSVC is fixed (i.e., $v = 3$).

for $\forall s_j \in \mathbb{S}$. The content of OSVC is updated for every v interval where v is also the size of OSVC. Specifically, Fig. 5 illustrates this behavior where at time $t + v$, the current OSVC content is erased and replaced with the loudest speakers ranging from $t + 1$ to $t + v$. ${}^{obs}\boldsymbol{\pi}(t)$, from 21 then is used to compare with a set of previously predicted state probability vectors to determine a suitable value for $c(t)$. With $c(t)$ known, a set of v predicted state probability vectors are computed over the timespan $t + 1, t + 2, \ldots t + v$ for each case of TMC size $c(t) - u(t)$, $c(t)$ and $c(t) + u(t)$. The predicted state probability vector at $t + n$ may be deduced from $\boldsymbol{\pi}(t)$ using the Chapman-Kolmogorov equations [37]

$$ {}^{c(t)-u(t)}\boldsymbol{\pi}(t+n) = \boldsymbol{\pi}(t)\left[{}^{c(t)-u(t)}\mathbf{P}(t)\right]^n \tag{22} $$

$$ {}^{c(t)}\boldsymbol{\pi}(t+n) = \boldsymbol{\pi}(t)\left[{}^{c(t)}\mathbf{P}(t)\right]^n \tag{23} $$

$$ {}^{c(t)+u(t)}\boldsymbol{\pi}(t+n) = \boldsymbol{\pi}(t)\left[{}^{c(t)+u(t)}\mathbf{P}(t)\right]^n \tag{24} $$

where $1 \leqslant n \leqslant v$ and $[.]^n$ is the transition matrix multiplied by itself n times. The output of (22)-(24) each represents a v dimensional vector respectively. For instance, if $v = 3$, the output of (22) represents three predicted state probability vectors, ${}^{c(t)-u(t)}\boldsymbol{\pi}(t+1)$, ${}^{c(t)-u(t)}\boldsymbol{\pi}(t+2)$, ${}^{c(t)-u(t)}\boldsymbol{\pi}(t+3)$. Therefore, an average of the predicted state probability vectors, is computed as

$$ {}^{q_l}\varphi_{s_j}(t) = \sum_{i \in \mathbb{I}_{OSVC}} {}^{q_l}\pi_{s_j}(t+i) \Bigg/ \sum_{i \in \mathbb{I}_{OSVC}, s_k \in \mathbb{S}} {}^{q_l}\pi_{s_k}(t+i) \tag{25} $$

where $q_l \in \mathbb{Q}(t)$ and $\mathbb{Q} = \{c(t) - u(t), c(t), c(t) + u(t)\}$. Equation (25) represents the average predicted state probabilities leading up to $t + v$. At time $t + v$, these vectors will then be compared with ${}^{obs}\boldsymbol{\pi}(t + v)$ such that

$$ d_l({}^{obs}\boldsymbol{\pi}(t+v), {}^{q_l}\varphi(t)) = \sqrt{\sum_{s_j \in \mathbb{S}} \left(\left({}^{q_l}\varphi_{s_j}(t) - {}^{obs}\pi_{s_j}(t+v)\right)^2 \times {}^{obs}\pi_{s_j}(t+v)\right)} \tag{26} $$

Based on d_l, $c(t+v)$ is determined to be

$$c(t+v) = q_{l^*} \tag{27}$$

where

$$l^* = \arg\min_{l}(d_l) \tag{28}$$

The size of the TMC is now set to $c(t+v)$ and will be used in constructing the transition matrices to determine dominant speakers $D_{t+v+1}, D_{t+v+2}, \ldots, D_{t+2v}$. The updated TMC_{size} remains fixed within the period of v seconds and via a similar process, will be updated again at $t+2v$, $t+3v$, etc.

6. Simulated Speaker Information Model

6.1. Generating Simulated Speaker Information

The augmented multi-party interaction (AMI) meeting corpus contains a multi-modal data set consisting of 100 hours of audio meeting recordings [21]. These recordings can be applied to evaluate the performance of a dominant speaker identification algorithm. However, the AMI recordings are limited to a four-point conference scenario. There are no existing publicly available meeting recordings for 5-point or larger conference scenario, but the proposed Markov chain algorithm is expected to work for a range of connected clients in a MVC session.

The process of creating a database of meeting recordings for a 5-point or larger conference scenario itself is a delicate task. It may require comprehensive test and/or role-playing cases under different conditions in order to yield the desired data sets. Even if such data sets were collected, ground truth references to evaluate and isolate transient speech patterns via manual labeling of the acoustic data risk high levels of variations between different labelers and it is deemed as problematic [1]. To minimize such variations, a high number of labelers would be required, to which resources for this task were not available at the time of this writing.

Therefore, this sub-section describes a method which generates a set of simulated loudest speaker information at every t second over a conference duration of one hour. The simulated speaker information represents either a 3, 4, 5, 7, 9 or 11-point conference scenario. The strategy here adopts a conversational pattern whereby if a current dominant speaker switches to a new client (e.g., Client A to Client B), the dominant speaker is likely to return back to previous speaker (i.e., Client A). This condition represents a form of loopback response (i.e., discussion) during a conversational process. The duration and likelihood occurrence of this loopback response is controlled using a threshold value and a pseudorandom token, which is generated for each t during which the simulated speaker information is populated.

Algorithm 1 describes the process of generating a set of simulated loudest speaker information. Each element in $TokArr[]$ refers to a set of threshold tokens based on the value of $|\mathbb{S}|$. For $|\mathbb{S}| = 3$, $TokArr[0] \leftarrow \{0, 70, 100\}$, for $|\mathbb{S}| = 4$,

Algorithm 1 Generating a set of simulated loudest speaker information

Input: $TokArr[]$:# token array
Input: $m \leftarrow 2$: # selected element in array
Initialize: $thld \leftarrow 90, y \leftarrow 0, t \leftarrow 0, j \leftarrow 0, l \leftarrow 3600, X_1 = s_0$
Output: $X_{t \leftarrow 1:l}$

```
1: for (t = 2 : l) do
2:     tok_0 ← rand()%100
3:     if (tok_0 ⩽ thld) then
4:         X_t = X_{t−1}, t++, y++
5:     else
6:         tok_1 ← rand()%100
7:         if (tok_1 ⩽ thld) then
8:             X_t = X_{(t−y)−1}
9:         else
10:            tok_2 ← rand()%100
11:            s_j = X_{t−1}, s_j ∈ 𝕊
12:            for (i = 1 : |𝕊| − 1) do
13:                if (tok_2 > TokArr[m][i − 1] && tok_2 ⩽ TokArr[m][i]) then
14:                    X_t = s_{(j+i)%|𝕊|} & exit for
15:                end if
16:            end for
17:        end if
18:    end if
19:    t++, y ← 0
20: end for
```

$TokArr[1] \leftarrow \{0, 70, 90, 100\}$ and so forth. These threshold tokens are used to determine a new loudest speaker at time t, based on the value of a randomly generated token. The value of m can be modified to reflect the selection of a different $\mathbb{S}$ content in generating the simulated speaker information.

In Algorithm 1, lines 1−18 computes X_t for $t = 2 : l\,(2, 3, 4, \cdots, l)$ with $l = 3600$ representing a one hour simulated conversation between $|\mathbb{S}|$ number of participants at one second intervals. In detail, line 2 assigns a pseudorandom number to the first token, tok_0. Lines 3−4 determine if X_t should take the form of X_{t-1} or otherwise. The concept here is to apply a 90% probability (i.e., $thld = 90$) of $X_t = X_{t-1}$, given that $0 < tok_0 \leqslant thld$. If $tok_0 > thld$, X_t no longer takes form of X_{t-1}, which invokes lines 6−18. Lines 7−8 determine if X_t should take the form of X_{t-y-1} or otherwise. A similar 90% probability is applied and compared to a second token, tok_1 whereby if $0 < tok_1 \leqslant thld$, $X_t = X_{(t-y)-1}$, which in turn represents a form of a loopback response during a conversation. Otherwise, Lines 10−14 are executed to randomly assign X_t such that $X_t \in \mathbb{S} \setminus \{X_{t-1}\}$

Fig. 6 illustrates the usage of Algorithm 1 in generating a sample set of simulated loudest speaker for $t = 1 : 20$s, with $|\mathbb{S}| = 5$. At $t = 1$s, X_1 is arbitrarily assigned to client A. The application of a random token is visible for $t > 1$s, where at $t = 2$s, given that $thld = 90$ and $tok_0 \leqslant thld$, $X_2 = X_1 = A$. This pattern continues up to $t = 8$s, where $tok_0 > thld$. This outcome indicates a switch in loudest speaker classification. tok_1 is generated and compared against $thld$ (see Line 5 in Algorithm 1) to determine if X_8 should take the form of

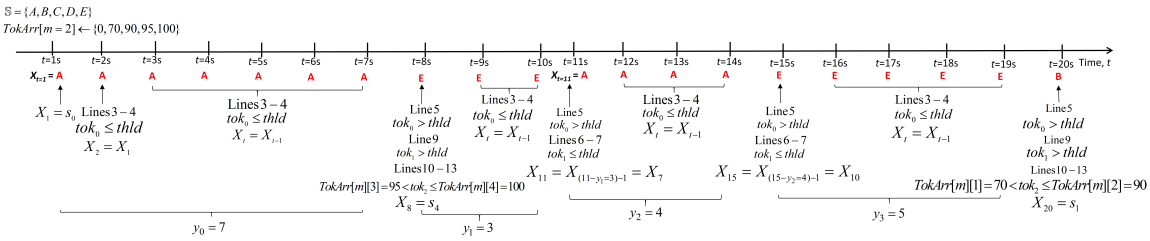


Figure 6: Sample population of simulated loudest speaker using Algorithm 1 for $t = 1 : 20$s, with $|\mathbb{S}| = 5$.

$X_{(8-y_0)-1}$ or otherwise. Here, $tok_1 > thld$ and thus Lines 9−13 are executed whereby tok_2 is generated and compared against each element in $TokArr[m]$, where $m = 2$. Given that $s_j = A$ at $t = 8$s and $j = 0$ (see line 13 in Algorithm 1), $TokArr[m][3] = 95 < tok_2 \leqslant TokArr[m][4] = 100$ where $i = 4$ and thus $X_8 = s_{(0+4)\%|\mathbb{S}|} = s_4 = E$.

At $t = 11$s, $tok_0 > thld$, which now indicates a second switch in loudest speaker classification. This time however, $tok_1 \leqslant thld$ and therefore $X_{11} = X_7 = A$, representing a simulated response from client A towards the previous conversation of client E. This pattern is repeated again at $t = 15$s, whereby $X_{15} = X_{10} = E$. At $t = 20$s, both tok_0 and tok_1 exceed $thld$, thus tok_2 is generated and compared against each element in $TokArr[m]$. Given that now $s_j = E$ at $t = 20$s and $j = 4$, $TokArr[m][1] = 70 < tok_2 \leqslant TokArr[m][2] = 90$ where $i = 2$ and thus $X_{20} = s_{(4+2)\%|\mathbb{S}|} = s_1 = B$.

6.2. Confidence Interval in Classifying Transient Speech Patterns

Sub-section A describes the methodology in generating a set of simulated speaker information. The outcome of Algorithm 1 exhibits different lengths of speech patterns, as illustrated in Fig. 6. In this figure, client E exhibits a smaller continuous speech length as the loudest speaker. Specifically, the transition of the loudest speaker from client A to E at $t = 8$s with a 3 second speech length would suggest that from $t = 8 : 10$s, client Es identification as the loudest speaker for this duration is in actuality a transient speech. However, the same might not be valid at $t = 15$s, where client E is again identified as the loudest speaker with a slightly longer 5 second speech length. In general, given the variability of speech characteristics in a MVC session, it would be infeasible to subjectively determine a specific length of transient speech. A more objective selection would be to analyze the speech length at each transition, in which a mean speech length, μ_s is computed as a measure of central tendency where

$$\mu_s = \frac{1}{|\mathbb{Y}|} \sum_{i \in \mathbb{Y}} y_i \tag{29}$$

In (29), $\mathbb{Y}$ represents a set of speech durations of a loudest speaker during a video communication session, with $y_i \in \mathbb{Y}$. For instance, in Fig. 6, $\mathbb{Y}$ would

contain elements $\{7, 3, 4, 5, 1\}$ with y_0 representing the speech duration of s_0 (i.e., client A) at $t = 1 : 7$s, y_1 representing the speech duration of s_5 (i.e., client E) at $t = 8 : 10$s and so forth. Speech lengths (i.e., y_i) that are smaller in value than μ_s are then classified as transient speech. However, the length of a MVC session is not fixed with durations ranging anywhere between a few minutes to several hours. Therefore in (29), μ_s represents a sample mean speech length and there is no measure of difference between μ_s and the population mean speech length, μ_p. As such, a 95% confidence interval for the mean is computed as an observed interval, which acts as a good estimate to the unknown μ_p.

The lower endpoint, λ of a confidence interval is computed as

$$\lambda = \bar{X} - \left(z \times \frac{\sigma_s}{\sqrt{|\overline{\mathbb{Y}}|}} \right) \tag{30}$$

where $\bar{X} = \mu_s$, σ_s represents the standard deviation of the sampled $\mathbb{Y}$ speech lengths and z represents the critical value obtained from the Standard Normal table. z is approximately 1.960 for a 95% confidence interval. λ as determined from (30) with $z = 1.96$ is the lowest possible estimate of μ_p at a 95% confidence interval. A reasonable threshold γ below which speech lengths y_i are considered transient speech may then be set as follows: Let y_{min} represent the smallest speech length of $\mathbb{Y}$, where $\forall y_i \in \mathbb{Y}$, $y_{\min} \geqslant y_i$. γ is then calculated as

$$\gamma = (y_{\min} + \lambda)/2. \tag{31}$$

If $y_i \leqslant \gamma$, y_i would now be classified as a transient speech length. Equation (31) is used to identify and substitute transient speech content in a dataset of simulated speaker information generated using Algorithm 1.

Algorithm 2 describes the process of classifying and substituting transient speech in simulated speaker data set. Lines 13 iterate each X_t element and compares it with X_{t-1}. If $X_t = X_{t-1}$, the y counter is increased. Subsequently when $X_t \neq X_{t-1}$, y is compared with γ and if $y \leqslant \gamma$, lines 67 substitute the content of X_t from $(t - y) + 1 : t$ with the previous non-transient speech data of X_{t-y}.

Fig. 7 revises the illustration of Fig. 6 by using Algorithm 2 to classify and substitute transient speech content. For $t = 8 : 10$s, $y_1 \leqslant \gamma$, which classifies this speech length as transient and substitutes the elements in this time range with that of $X_t = X_{t-y_1}$. A similar pattern is also observed for $t = 11 : 14$s. For $t = 1 : 7$s and $t = 15 : 19$s, both y_0 and y_3 are larger than γ, thus the speaker information is left unmodified.

Recall from Section 4 that after every v interval, a decision has to be made on whether to increase TMC_{size} to $c(t)+u(t)$, maintain TMC_{size} at $c(t)$ or reduce TMC_{size} to $c(t) - u(t)$. $c(t)$ is the last TMC_{size} applied and its determination has been expounded in the same section. The determination of $u(t)$ is expounded here. $u(t)$ is the amount by which TMC_{size} is incremented or decremented. Similar to $c(t)$, $u(t)$ is periodically updated based on the value of γ at time t as

Algorithm 2 Classification and substitution of transient speech data

Input: γ using (31)
Initialize: $l \leftarrow 3600, y \leftarrow 0$
Output: $X_{t \leftarrow 1:l}$

```
    for (t = 2 : l) do
 2:     if (X_t = X_{t-1}) then
            y ← y + 1
 4:     else
            if (y ⩽ γ) then
 6:             for (j = (t − y) + 1 : t do
                    X_j ← X_{t−y}
 8:             end for
            end if
10:         y ← 0
        end if
12: end for
```

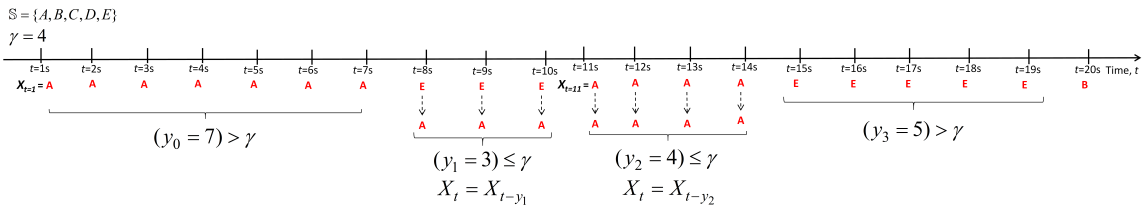


Figure 7: Sample classification and substitution of transient speech data using Algorithm 2, assuming $\gamma = 4$.

follows:

$$u(t) = \gamma(t) = (y_{\min} + \lambda(t))/2 \tag{32}$$

where

$$\lambda(t) = \bar{X} - \left(z \times \frac{\sigma_s(t)}{\sqrt{|\mathbb{Y}_t|}} \right) \tag{33}$$

Equations (32) and (33) are similar to (30) and (31), except that λ and γ now vary with time. $\mathbb{Y}_t$ in (33) is the set of speech durations of loudest speakers from time 0 to t and $\sigma_s(t)$ is the standard deviation of elements in $\mathbb{Y}_t$. Obviously, $\lim_{t \to \infty} \mathbb{Y}_t = \mathbb{Y}$. With (32), the reduction or expansion of the TMC would either remove or include significant older speech patterns in computing the predicted state probability vectors. This in turn potentially improves both the level of accuracy in predicting a dominant speaker and reduces misclassification of a dominant speaker.

7. Performance Assessment

The performance of the Markov chain algorithm and the proposed enhancement were assessed based on the following criteria: a) Level of accuracy in pre-

dicting a dominant speaker for a given set of speech information; b) Reduction of dominant speaker misclassification (i.e., Transient reduction rate) due to transient speech. To recap, the dominant speaker identification algorithm proposed by Volfin and Cohen [33] was assessed using simulated concatenated speech segments from the TIMIT database [12]. However, the applied 1s boundary for both long term audio properties and tolerable transition delay differs from the objectives of this paper, which analyzes speech activities beyond a 1s duration and conversational patterns in identifying a dominant speaker. Therefore, the Markov chain algorithm and the proposed enhancement were benchmarked against a straightforward approach using a basic SVC, which was described in Section 3. Each algorithm was analyzed for its accuracy in correctly identifying dominant speakers in the presence of transient speech patterns. Table 3 lists the terminologies assigned to the analyzed dominant speaker identification algorithms throughout this section. All of the tested algorithms were developed using Visual C++ into a complete software implementation.

7.1. Performance Assessment using Simulated Speaker Information

Table 3: Terminology of the dominant speaker identification algorithms

Terminology	Description	Settings
B-SVC	Straightforward dominant speaker identification algorithm using a basic SVC.	$SVC_{size} = 5$
MC-Const	Markov chain algorithm with constant weights for SVC and TMC, with a dynamic TMC size.	Initial $TMC_{size} = 50$ $20 \leq TMC_{size} \leq 200$ $SVC_{size} = 20$ $OSVC_{size} = 3$
MC-Lin	Markov chain algorithm with linear weights for SVC and TMC, with a dynamic TMC size.	Initial $TMC_{size} = 50$ $20 \leq TMC_{size} \leq 200$ $SVC_{size} = 20$ $OSVC_{size} = 3$
MC-NonLinPos	Markov chain algorithm with non-linear weights ($\alpha > 0$) for SVC and TMC, with a dynamic TMC. size.	Initial $TMC_{size} = 50$ $20 \leq TMC_{size} \leq 200$ $SVC_{size} = 20$ $OSVC_{size} = 3$ $\alpha = 0.9$
MC-NonLinNeg	Markov chain algorithm with non-linear weights ($\alpha < 0$) for SVC and TMC, with a dynamic TMC. size.	Initial $TMC_{size} = 50$ $20 \leq TMC_{size} \leq 200$ $SVC_{size} = 20$ $OSVC_{size} = 3$ $\alpha = -0.9$

Using Algorithm 1 and for each case of 3, 4, 5, 7, 9 and 11 clients, a set of 50 simulated loudest speaker information were generated, and the average results were then compiled for analysis. Fig. 8 revises Fig. 7 to illustrate a sample computation of the prediction accuracy and transient speech reduction rate. Each generated data set using Algorithm 1 (i.e., $data\_x$ (where $x = 1 : 50$)) represents a one-dimensional array with l number of X_t elements (i.e., $t = 1 : l$, where $l = 3600$s). Each X_t element was computed at a one second interval, hence

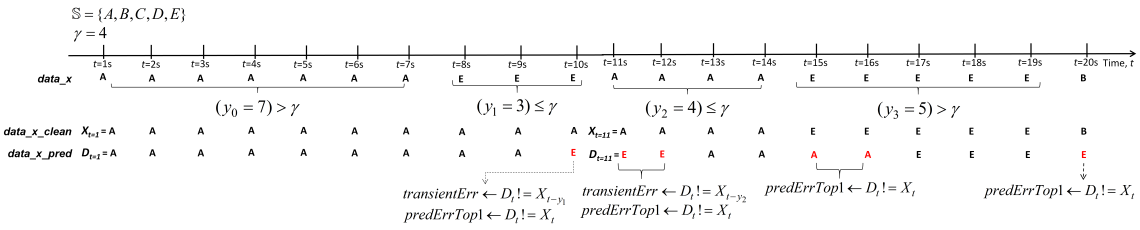


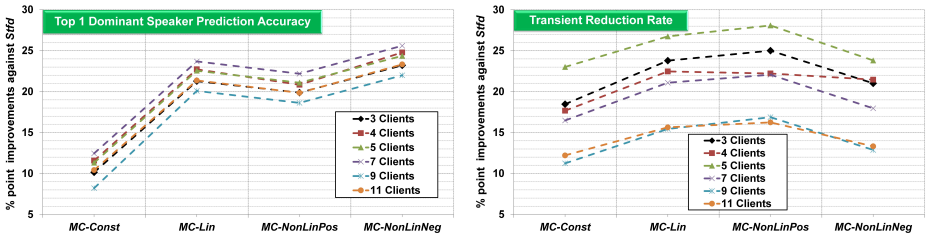
Figure 8: Sample computation of the prediction error (i.e., $predErrTop1$) and transient mispredictions (i.e., $transientErr$).

constituting a one hour simulated conversation. Each data set also includes randomly generated transient speech patterns.

The content of *data_x* were then analyzed using a dominant speaker identification algorithm, which in turn generated a two-dimensional predicted data set (i.e., *data_x_pred*). *data_x_pred* contains a pair of speakers where the first column represents the speaker, which statistically has the highest probability of being the dominant speaker at time t, D_t (i.e., most dominant speaker). The second column represents the speaker with the second highest probability of being the dominant speaker at time t (i.e., second most dominant speaker). For the purpose of brevity, the second column is not illustrated in Fig. 8. Note that the content of *data_x* were also utilized by Algorithm 2 to classify and substitute transient speech content, which in turn generates a transient free speech (i.e., cleaned) data set, *data_x_clean*. The content of *data_x*, *data_x_clean* and *data_x_pred* are then analyzed to determine the level of accuracy in predicting a dominant speaker and the transient reduction rate. The aim here is for the content of *data_x_pred* to closely match *data_x_clean*.

In analyzing the sample illustration of Fig. 8, for $t = 8:10$s, $y_1 \leqslant \gamma$, which in turn classifies this speech length as transient. Therefore *data_x_clean* substitutes the transient speech (i.e., client E) with $X_t = X_{t-y_1}$ (i.e., client A). The applied dominant speaker identification algorithm for *data_x_pred* is able to accurately predict $D_t = A$ for $t = 8:9$s. However at $t = 10$s, D_{10} is inaccurately predicted as client E, which adds to both the transient mispredictions (i.e., $transientErr$) and top 1 dominant speaker prediction error (i.e., $predErrTop1$). A similar pattern is also illustrated for $t = 11:12$s. For $t = 15:19$s, $y_3 > \gamma$, which classify this speech length as non-transient. However, a two second delay is observed from $t = 15:16$s before D_t is accurately predicted as client E. This delay is categorized as a prediction error (i.e., $D_{15:16}! = X_{15:16}$) and added into $predErrTop1$. This delay is also observed at $t = 20$s. The prediction accuracy analysis for the top 2 dominant speakers is similar to that of the top 1 dominant speaker, and hence is not illustrated here.

Note that the computed mispredictions, $predErrTop1$, considers both the errors caused by delays in transitioning from one dominant speaker to another and misclassification of a dominant speaker during transient speech periods. Conversely, the computed $transientErr$ only factors in errors caused by mis-



(a) (b)

Figure 9: Percentage point improvements in (a) top 1 dominant speaker prediction accuracy and (b) transient reduction rate of *MC-Const*, *MC-Lin*, *MC-NonLinPos* and *MC-NonLinNeg* against *B-SVC*, using the simulated speaker information.

classification of a dominant speaker during transient speech periods. A low $predErrTop1$ does not necessarily imply that a dominant speaker identification algorithm is able to minimize the impact of transient speech in misclassification of dominant speakers. Instead, a balance is required whereby a good prediction accuracy is complimented with an equally robust transient reduction rate.

Table 4 tabulates results of the prediction accuracy and the transient reduction rate for the analyzed algorithms in Table 3, based on varying number of conference clients (i.e., 3, 4, 5, 7, 9 and 11 clients). Since each assessment consists of 50 data sets, an average of the prediction accuracy and transient reduction rate were tabulated in Table 4. Fig. 9(a) illustrates the percentage point improvements in top 1 dominant speaker prediction accuracy of *MC-Const*, *MC-Lin*, *MC-NonLinPos* and *MC-NonLinNeg* algorithms against that of the benchmark *B-SVC* algorithm. Fig. 9(b) depicts a similar illustration for the transient reduction rate.

Table 5 tabulates the average results of the percentage point improvements as illustrated in Fig. 9 (of all clients) for both top 1 dominant speaker prediction accuracy and transient reduction rate. In analyzing the performance of the benchmark *B-SVC* algorithm in Table 4, this algorithm correctly predicted the top 1 dominant speaker at a rate of between 58% and 60% for 3, 4, 5, 7, 9 and 11 clients. Top 2 dominant speakers were accurately predicted at a higher rate of between 93% and 96%, and misclassification of dominant speakers (i.e., Transient reduction rate) were reduced at a rate of between 63% and 65%. Recall that the aim of the Markov chain algorithm is to increase the reliability in accurately predicting a dominant speaker and results of Table 4 and Fig. 9 concurs with these objectives. In detail, the *MC-Const* registered a higher top 1 dominant speaker prediction accuracy at a rate of about 71%, with an average 12.81 percentage point (pp) increase against *B-SVC* (see Table 5). Additionally, *MC-Const* also registered higher reductions in misclassification of dominant speakers at a rate of between 80% and 82%, with an average 17 pp increase against *B-SVC*.

Using linear weights, *MC-Lin* validates the hypothesis of Section 4, whereby

Table 4: Prediction accuracy and transient reduction rate of (a) *B-SVC*, *MC-Const*, *MC-Lin* and (b) *MC-NonLinPos* and *MS-NonLinNeg* for 3, 4, 5, 7, 9 and 11 clients

(a)

	B-SVC			*MC-Const*			*MC-Lin*		
$\mathbb{S}$	Top 1 Dominant Speaker Accuracy (%)	Top 2 Dominant Speaker Accuracy (%)	Transient Reduction Rate (%)	Top 1 Dominant Speaker Accuracy (%)	Top 2 Dominant Speaker Accuracy (%)	Transient Reduction Rate (%)	Top 1 Dominant Speaker Accuracy (%)	Top 2 Dominant Speaker Accuracy (%)	Transient Reduction Rate (%)
3	59.82	96.18	65.30	71.71	99.11	82.37	82.87	99.66	85.80
4	59.11	94.83	65.49	71.51	98.84	81.19	82.68	99.58	84.81
5	58.58	94.08	64.70	71.75	98.63	82.48	82.93	99.46	85.18
7	58.72	93.97	63.53	71.40	98.61	80.70	82.73	99.45	84.39
9	58.15	93.52	64.13	71.47	98.64	82.00	82.91	99.43	84.95
11	58.16	93.75	65.42	71.56	98.64	82.32	82.91	99.48	85.57

(b)

	MC-NonLinPos			*MC-NonLinNeg*		
$\mathbb{S}$	Top 1 Dominant Speaker Accuracy (%)	Top 2 Dominant Speaker Accuracy (%)	Transient Reduction Rate (%)	Top 1 Dominant Speaker Accuracy (%)	Top 2 Dominant Speaker Accuracy (%)	Transient Reduction Rate (%)
3	81.19	99.61	86.88	84.67	99.70	83.06
4	81.03	99.51	85.97	84.43	99.63	82.10
5	81.27	99.40	86.44	84.66	99.52	82.32
7	81.05	99.39	85.59	84.52	99.51	81.52
9	81.24	99.36	86.25	84.62	99.49	82.08
11	81.21	99.41	86.79	84.64	99.53	82.42

Table 5: Average percentage point improvements (of all clients) for both top 1 dominant speaker prediction accuracy and transient reduction rate of the Markov Chain algorithms against *B-SVC*

Algorithm	Average improvements in top 1 dominant speaker prediction accuracy	Average improvements in transient reduction rate
MC-Const	12.81 pp	17.08 pp
MC-Lin	24.08 pp	20.35 pp
MC-NonLinPos	22.41 pp	21.56 pp
MC-NonLinNeg	25.83 pp	18.48 pp

in Tables 4 and 5 and in Fig. 9, the level of accuracy in predicting the most dominant speaker substantially increased at a rate of about 82%, and with an average 24 pp increase against *B-SVC*. In addition, reductions in misclassification of dominant speakers notably improved to a high of 86% with an average 20.35 pp increase against *B-SVC*.

Using non linear weights, *MC-NonLinPos* registered an average 22.41 pp

increase against *B-SVC* in in top 1 prediction accuracy (see Table 5), which is about 2 points lower than *MC-Lin*s average improvements against *B-SVC*. This dip is attributed towards a positive exponential growth constant (i.e., $\alpha = 0.9$) in *MC-NonLinPos*, which corresponds to a slower response identifying the change in dominant speaker. Nonetheless, a positive α in *MC-NonLinPos* marginally improves reductions in misclassifications of a dominant speaker with an average 21.56 pp increase against B-SVC, which is about 1.2 points higher than *MC-Lin*s average improvements against *B-SVC*. *MC-NonLinNeg* registered the highest performance in top 1 dominant speaker prediction accuracy at 84%, and with an average 25.83 pp increase against *B-SVC*. The negative exponential growth constant (i.e., $\alpha = -0.9$) in *MC-NonLinNeg* translates into a faster response in identifying changes between dominant speakers, albeit at the expense of higher susceptibility towards transient speech patterns. Consequently, *MC-NonLinNeg* registered an average 18.48 pp increase against *B-SVC* in transient reduction rates, which is about 3 points lower than *MC-NonLinPos*s average improvements against *B-SVC*.

7.2. Performance Assessment using AMI Meeting Corpus Speaker Data

The preceding sub-section assessed the Markov chain algorithm and the proposed enhancement against the benchmark *B-SVC* algorithm using a set of simulated speaker information. Results from this assessment demonstrate substantial improvements in both the level of accuracy in predicting a dominant speaker and transient reduction rates.

To further validate the performance of the aforementioned algorithms, this subsection analyses the performance of these algorithms by using a set of recorded meeting conversations from the AMI Meeting Corpus database. These meeting conversations were recorded as face-to-face conversations between a maximum of four individuals in a single room, which was setup for both close-talking and far-field audio recording. A face-to-face conversation would typically indicate a quicker conversational response than that of teleconference conversation. Nevertheless, AMI meeting conversations represent a collection of scenario and non-scenario based conversations, which includes speech patterns that can be objectively classified as transient using Algorithm 2.

In addition, the AMI database provides classification of a recorded meeting conversation into individual headsets (a maximum of four). These individual recordings can then be used to emulate a teleconference conversation. Hence, a sample of three non-scenario based: *EN2002a* ($l = 2142$s), *EN2006a* ($l = 3525$s) and *EN2006b* ($l = 3052$s), and three scenario based: *ES2006a* ($l = 1284$s), *ES2013c* ($l = 2358$s) and *ES2016c* ($l = 2308$s) meeting recordings were applied for this analysis. The scenario based meetings are controlled based meetings with specific individual functions and goals. The non-scenario meetings are naturally occurring meetings in a variety of modes. As each recorded meeting is restricted to a maximum of four participants, $|\mathbb{S}|$ is fixed at four for the rest of this sub-section. A similar methodology as described in the preceding sub-section was applied to compute the prediction accuracy and transient rate reduction of the algorithms in Table 3 using the sample AMI meeting recordings.

Since the AMI meeting recordings were used instead of a simulated speaker information, Algorithm 1 is not applied in this sub-section. Instead, (1) was applied for each sampled AMI waveform audio recording to compute X_t for $t = 1 : l$.

The AMI meeting recording data were then channeled into Algorithm 2 to classify and substitute transient speech content, which in turn generated a transient free speech data set, (e.g., $EN2002a\_clean$). Each recording was also fed into the analyzed dominant speaker identification algorithms, which generates a predicted two-dimensional data set (e.g., $EN2002a\_pred$), with a similar content layout as described in the preceding sub-section (See Fig. 8). The content of the AMI meeting recording data, cleaned data and predicted data were then analyzed to determine the level of accuracy in predicting a dominant speaker and the transient reduction rate.

Table 6: Prediction accuracy and transient reduction rate assessment of (a) *B-SVC*, *MC-Const*, *MC-Lin* and (b) *MC-NonLinPos* and *MS-NonLinNeg*, for EN2002a, EN2006a, EN2006b, ES2006a, ES2013c and ES2016c

(a)

	B-SVC			*MC-Const*			*MC-Lin*		
$\mathbb{S} = 4$	Top 1 Dominant Speaker Accuracy (%)	Top 2 Dominant Speaker Accuracy (%)	Transient Reduction Rate (%)	Top 1 Dominant Speaker Accuracy (%)	Top 2 Dominant Speaker Accuracy (%)	Transient Reduction Rate (%)	Top 1 Dominant Speaker Accuracy (%)	Top 2 Dominant Speaker Accuracy (%)	Transient Reduction Rate (%)
EN2006a	63.15	86.57	66.87	67.95	92.61	72.15	77.59	96.78	80.55
EN2006b	77.72	93.71	76.99	77.76	96.84	79.63	82.15	98.60	81.26
EN2002a	49.74	78.36	50.00	64.84	91.73	70.38	78.02	96.70	82.34
ES2006a	71.90	91.74	57.78	75.71	94.90	62.22	85.34	98.46	80.00
ES2013c	74.49	90.52	54.85	82.03	95.11	75.75	89.43	97.88	83.21
ES2016c	83.80	95.53	73.64	89.15	97.52	86.36	93.09	98.89	82.73

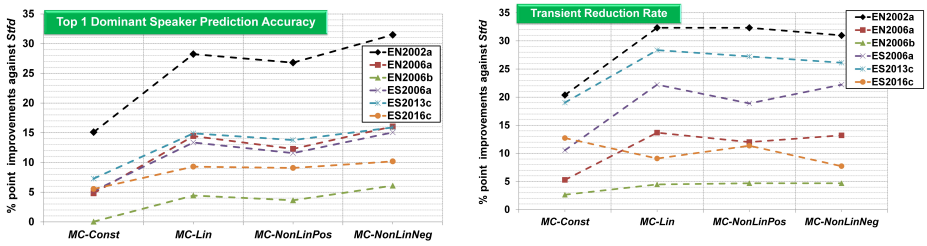
(b)

	MC-NonLinPos			*MC-NonLinNeg*		
$\mathbb{S} = 4$	Top 1 Dominant Speaker Accuracy (%)	Top 2 Dominant Speaker Accuracy (%)	Transient Reduction Rate (%)	Top 1 Dominant Speaker Accuracy (%)	Top 2 Dominant Speaker Accuracy (%)	Transient Reduction Rate (%)
EN2006a	75.43	96.03	78.87	79.20	96.89	80.07
EN2006b	81.29	98.30	81.67	83.88	98.87	81.67
EN2002a	76.45	96.61	82.34	81.13	96.99	80.98
ES2006a	83.64	98.22	77.22	87.04	98.46	80.00
ES2013c	88.22	97.57	82.09	90.39	97.92	80.97
ES2016c	92.83	98.80	85.00	93.98	98.80	81.36

Table 6 compiles results of the prediction accuracy and the transient reduction rate for the analyzed algorithms in Table 3, based on the sampled

Table 7: Average percentage point improvements (AMI Recordings) for both top 1 dominant speaker prediction accuracy and transient reduction rate of the Markov Chain algorithms against *B-SVC*

Algorithm	Average improvements in top 1 dominant speaker prediction accuracy	Average improvements in transient reduction rate
MC-Const	6.32 pp	12.01 pp
MC-Lin	14.11 pp	18.36 pp
MC-NonLinPos	12.86 pp	17.76 pp
MC-NonLinNeg	16.80 pp	17.50 pp



(a) (b)

Figure 10: Percentage point improvements in (a) top 1 dominant speaker prediction accuracy and (b) transient reduction rate of *MC-Const*, *MC-Lin*, *MC-NonLinPos* and *MC-NonLinNeg* against *B-SVC*, using sample recordings from the AMI Meeting Corpus database.

AMI meeting recordings. Fig. 10(a) illustrates the percentage point improvements in top 1 dominant speaker prediction accuracy of *MC-Const*, *MC-Lin*, *MC-NonLinPos* and *MC-NonLinNeg* algorithms against that of the benchmark *B-SVC* algorithm. Fig. 10(b) depicts a similar illustration for the transient reduction rate. Table 7 tabulates the averages results of the percentage point improvements as illustrated in Fig. 10 (of all sample AMI meeting recordings) for both top 1 dominant speaker prediction accuracy and transient reduction rate.

In analysing the performance of the benchmark *B-SVC* algorithm in Table 6, this algorithm accurately predicted the top 1 dominant speaker at a rate between 49% and 84% for the sampled AMI meeting recordings. Top 2 dominant speakers were accurately predicted at a higher rate between 78% and 95%. In addition, transient reduction were registered at a rate between 50% and 77%. The Markov chain algorithms registered significant improvements in the prediction accuracy and higher reductions in misclassification of dominant speakers. Specifically, *MC-Const* registered a higher top 1 prediction accuracy between 67% and 89% for the sampled AMI meeting recordings. This constitutes an average 6.32 pp increase in prediction accuracy against *B-SVC* (see Table 7). Additionally, *MC-Const* also registered higher transient reduction at a rate of between 70% and 86%, which translates into an average 12 pp increase against *B-SVC*.

Similarly, the usage of Markov chain with weight coefficients (linear and non

linear) for the sampled AMI meeting recordings also registered substantial improvements in both prediction accuracy and transient rate reduction. In detail, *MC-Lin* increases top 1 prediction accuracy at a rate of between 77% and 93% with an average 14.11 pp increase against *B-SVC*. In addition, reductions in misclassification of dominant speakers notably improved to a high of 82.7% for *ES2016c* with an overall average 18.36 pp increase against *B-SVC*. A similar dip in the top 1 dominant speaker predicition accuracy as previously observed in Table 4 is also observed in Table 6 for the *MC-NonLinPos* algorithm, with a smaller average increase rate of 12.86 pp against *B-SVC*. To reiterate, this dip is attributed towards in *MC-NonLinPos*, which corresponds to a slower response in identifying changes in a dominant speaker. The *MC-NonLinNeg* algorithm registered the highest performance in top 1 dominant speaker prediction accuracy with results ranging between 79% and 93%, and with an average 16.8 pp increase against *B-SVC*. This algorithm also registered high transient reduction rates with an average 17.50 pp increase against *B-SVC*.

8. Conclusion

8.1. Summary of Contributions

A discrete-time Markov chain algorithm is proposed to accurately predict a dominant speaker in a multipoint video communication session. The proposed method here addresses the impact of variability in speech characteristics due to transient speech or noise patterns during a MVC session. The proposed method applies a transition probability matrix which statistically defines the probabilities of speech transitions from one speaker to another speaker. Coupled with an initial state vector, the statistical properties of the system's future is predicted, in which these properties are computed and used to identify a dominant speaker. To enhance the responsiveness of this algorithm towards changes in dominant speakers, a set of state and transition weights were proposed for both SVC and TMC respectively.

In addition, the variability of speech characteristics during a video communication session necessitates the need to dynamically resize the transition probability matrix container (i.e., TMC). Hence, an observed state vector is periodically compared with a set of previously predicted state probability vectors for a reduced, maintained and expanded TMC. The predicted state probability vector that closely matches the observed state vector defines the revised size of the TMC. The Markov chain algorithms with a dynamically resized TMC were assessed using a set of simulated speaker information and a set of sampled AMI meeting recordings. Results from these assessments demonstrates that the proposed enhanced Markov chain algorithm exhibits significant improvements in prediction accuracies and transient reduction rates against a benchmarked basic state vector approach.

Results in Fig. 9 suggest a closer results correlation between the tested number of simulated conference clients (i.e., 3, 4, 5, 7, 9 and 11 clients). This correlation is indeed lower in 10. These differences are due to the lack of natural

cues in the artificially simulated (generated) speaker data set. Reason being is that the simulated speaker data algorithm (i.e., Algorithm 1) focuses on incorporating loopback responses between clients in a typical video communication session. This algorithm does not factor in natural cues in generating the artificial loudest speech pattern. Due to the lack of resources to record speech data for large numbers of conference clients, the idea of Algorithm 1 is to generate an artificial loudest speaker data set for large numbers of conference clients to observe the level of accuracy and transient reduction using both the proposed Markov chain and benchmark dominant speaker detection algorithms.

Relying purely on an artificial speaker data set to measure the reliability of the proposed Markov chain algorithm is insufficient. As such, sub section 7.2 utilizes a sample of recorded speech data from the AMI meeting corpus with each recording consisting of a maximum of four endpoint clients. On top of the loopback responses, the AMI meeting recordings also include varying natural cues for each recording, which are not visible in Algorithm 1. Hence, the percentage point improvements using the AMI recordings in 10 exhibits smaller levels of correlation to that of 9.

8.2. Future Work

A Markov chain captures the next simplest sort of dependence where the probability distribution of a next state depends only on the current state. Hence, for future work, a higher amount of memory could be included into these states by using a higher order Markov chain. A higher order Markov chain could potentially improve on the statistical analysis of conversational patterns in a video communication session. This in turn would yield a faster transition time for genuine dominant speaker, which could improve both prediction accuracies and transient reduction rates.

In addition, the proposed dynamic transition window algorithm was designed to adaptively adjust the value of TMC_{size}, within a range of between 20 and 200 elements which was defined during performance assessment. This algorithm could be expanded to adaptively adjust the value of SVC_{size}. However, by doing so, this process would now require generating an additional three sets of state probability vectors for a reduced, maintained and expanded SVC. Then, each state probability vector would be applied into (22) - (26) to compute the distance between the observed and state probability vectors for a reduced, maintained and expanded TMC. These additional procedures increases the computational complexities of the dynamic transition window algorithm, to which the impact of this method on the prediction accuracy and reduction in dominant speaker misclassification could indeed be further explored. Methods to dynamically modify the values of $OSVC_{size}$ and the exponential decay constant (i.e., α) could also be beneficial in improving both the prediction accuracies and transient reduction rates.

Apart form enhancements to the Markov chain algorithm, natural cues including voice intonation, special words and speech expressions were not applied in this paper as part of detecting a dominant speaker or a switch between speakers. The reason being is that such methods would require a deeper speech

context analysis of each conference endpoint client, which in turn could also risk being computationally expensive as a real-time system for a large number of conference participants. Nevertheless, these techniques could be indeed be explored to compliment the proposed solution here.

References

[1] Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., Vinyals, O., 2 2012. Speaker diarization: A review of recent research. IEEE Transactions on Audio, Speech, Language Process 20 (2), 356–370.

[2] Baskaran, V. M., Chang, Y. C., Loo, J., Wong, K., Gan, M.-T., 2015. Dominant speaker detection using discrete markov chain for multi-user video conferencing. In: Proc. IEEE ICCE-TW. Taipei, pp. 492–493.

[3] Baskaran, V. M., Wong, K., 2010. Audio mixer with automatic gain controller for software based multipoint control unit. In: Proc. APCCAS. KL, pp. 164–167.

[4] Bauer, S., Clark, D., Lehr, W., 2009. The evolution of internet congestion. In: Proc. TPRC. Arlington, VA, pp. 1–34.

[5] Chetty, M., Banks, R., Brush, A. J. B., Donner, J., Grintter, R. E., 2012. You're capped: Understanding the effects of bandwidth caps on broadband use in the home. In: Proc. CHI. TX, pp. 3021–3030.

[6] Dove, D., Talmon, R., Cohen, I., 4 2015. Audio-visual voice activity detection using diffusion maps. IEEE/ACM Transactions on Audio, Speech, and Language Processing 23 (4), 732–745.

[7] Dove, D., Talmon, R., Cohen, I., 12 2016. Kernel method for voice activity detection in the presence of transients. IEEE/ACM Transactions on Audio, Speech, and Language Processing 24 (12), 2313–2326.

[8] Fang, W., Yin, X., An, Y., Xiong, N., Guo, Q., Li, J., 4 2015. Optimal scheduling for data transmission between mobile devices and cloud. Information Sciences 301, 169–180.

[9] Fapi, T., Rossignol, E., Eric, P., 2014. Selection of active speaker(s) in voip conference bridges: From linear domain to celp parameters domain. In: Proc. IEEE Region 10 Symposium. KL, pp. 466–470.

[10] Firestone, S., Ramalingam, T., Fry, S., 3 2007. Voice and video conferencing fundamentals, 1st Edition. Cisco Press, Indiana.

[11] Fung, K. T., Chan, Y. L., Siu, W.-C., 2 2004. Low-complexity and high-quality frame-skipping transcoder for continuous presence multipoint video conferencing. IEEE Transactions on Multimedia 6 (1), 31–46.

[12] Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., Dahlgren, N. L., Zue, V., 1993. The Linguistic Data Consortium Catalog. The Linguistic Data Consortium, Philadelphia, Ch. TIMIT acoustic-phonetic continuous speech corpus.

[13] Hung, H., Huang, Y., Friedland, G., Gatica-Perez, D., 2008. Estimating the dominant person in multi-party conversations using speaker diarization strategies. In: Proc. ICASSP. Las Vegas, pp. 2197–2200.

[14] Hung, H., Jayagopi, D., Yeo, C., Friedland, G., Ba, S., 2007. Using audio and video features to classify the most dominant person in a group meeting. In: Proc. of ACM Multimedia. Germany, pp. 835–838.

[15] Jana, S., Pande, A., Chan, A., Mohapatra, P., 6 2013. Mobile video chat: issues and challenges. IEEE Communication Magazine 51 (6), 144–151.

[16] Jayagopi, D., Hung, H., Yeo, C., Gatica-Perez, D., 3 2009. Modeling dominance in group conversations from non-verbal activity cues. IEEE Transactions on Audio, Speech, Language Process 17 (3), 501–513.

[17] Jie, C., Peng, P., 2010. Recognize the most dominant person in multi-party meetings using nontraditional features. In: Proc. IEEE Int. Conf. on Intell. Comput. and Intell. Syst. Xiamen, pp. 312–316.

[18] Lee, J., Lee, K., Han, C., Kim, T., Chong, S., 12 2016. Resource-efficient mobile multimedia streaming with adaptive network selection. IEEE Transactions on Multimedia 18 (12), 2517–2527.

[19] Lin, C.-W., Chen, Y.-C., Sun, M. T., 10 2003. Dynamic region of interest transcoding for multipoint video conferencing. IEEE Transactions on Circuits and Systems for Video Technology 13 (10), 982–992.

[20] Mast, M. S., 7 2002. Dominance as expressed and inferred through speaking time. Human Communication Research 28 (3), 420–450.

[21] McCowan, I., et al., 2005. The ami meeting corpus. In: Proc. Meas. Behavior.

[22] Nagata, Y., Fujioka, T., Abe, M., 1 2006. Speech enhancement based on auto gain control. IEEE Transactions on Audio, Speech and Language Processing 14 (1), 177–190.

[23] Qi, X., Yang, Q., Nguyen, D. T., Peng, G., Zhou, G., Dai, B., Zhang, D., Li, Y., 8 2016. A context-aware framework for reducing bandwidth usage of mobile video chats. IEEE Transactions on Multimedia 18 (8), 1640–1649.

[24] Ramrez, J., Grriz, J. M., Segura, J. C., 6 2007. Robust Speech Recognition and Understanding. I-Tech Education and Publishing, Vienna, Ch. Voice activity detection. Fundamentals and speech recognition system robustness, pp. 1–22.

[25] Ramrez, J., Segura, J. C., Bentez, C., Garcia, L., Rubio, A., 9 2005. Statistical voice activity detection using a multiple observation likelihood ratio test. IEEE Signal Processing Letters 12 (10), 689–692.

[26] Ramrez, J., Segura, J. C., Bentez, C., Torre, A. D.-L., Rubio, A., 4 2004. Efficient voice activity detection algorithms using long-term speech information. Speech Communication 42 (3–4), 271–287.

[27] Rienks, R., Heylen, D., 7 2005. Dominance detection in meetings using easily obtainable features. Machine Learning for Multimodal Interaction, Lecture Notes in Computer Science 3869, 76–86.

[28] Rienks, R., Zhang, D., Gatica-Perez, D., Post, W., 2006. Detection and application of influence rankings in small group meetings. In: Proc. ICMI '06. NY, pp. 257–264.

[29] Sheikh, H. R., Liu, S., Zhou, W., Bovik, A. C., 2002. Foveated multipoint videoconferencing at low bit rates. In: Proc. ICASSP. Orlando, pp. II–2069–II–2072.

[30] Sohn, J., Kim, N. S., Sung, W., 1 1999. A statistical model-based voice activity detection. IEEE Signal Processing Letters 6 (1), 1–3.

[31] Sun, M. T., Loui, A. C., Chen, T. C., 12 1997. A coded-domain video combiner for multipoint continuous presence video conferencing. IEEE Transactions on Circuits and Systems for Video Technology 7 (6), 855–863.

[32] Sun, M. T., Wu, T.-D., Hwang, J.-N., 5 1998. Dynamic bit allocation in video combining for multipoint conferencing. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 45 (5), 644–648.

[33] Volfin, I., Cohen, I., 6 2013. Dominant speaker identification for multipoint videoconferencing. Computer Speech and Language 27 (4), 895–910.

[34] Xing, F., Wei-kang, G., Xiu-qing, Y., 6 2005. Research on fast real-time adaptive audio mixing in multimedia conference. Journal of Zhejiang University Science A 6 (6), 507–512.

[35] Xu, X., He, L.-W., Florencio, D., Rui, Y., 2006. Pass: peer-aware silence suppression for internet voice conference. In: Proc. IEEE ICME. Toronto, pp. 2149–2152.

[36] Yang., M., Groves, T., Zheng, N., Cosman, P., 11 2014. Iterative pricing-based rate allocation for video streams with fluctuating bandwidth availability. IEEE Transactions on Multimedia 16 (7), 1849–1862.

[37] Yates, R. D., Goodman, D. J., 2005. Probability and stochastic processes. A friendly introduction for electrical and computer engineers, 2nd Edition. John Wiley and Sons, USA, Ch. Markov chains, pp. 445–500.