A General Approach to Classifying Mode of Speech: The Speaker-Affect Model for Audio Data*

Dean Knox† Christopher Lucas‡

First draft: August 28, 2015
This draft: April 2, 2017

Abstract

Though we generally assume otherwise, humans communicate using more than bags of words alone. Auditory cues convey important information, such as emotion, in many contexts of interest to political scientists. However, in part due to the relative difficulty of processing and analyzing audio data, research has disproportionately focused on the textual component of pre-transcribed corpora. To resolve this, we develop a general approach, the Speaker-Affect Model (SAM), capable of classifying speech into user-specified "modes," which can be emotional content, speaker IDs, or any other set of labels with distinct audio profiles. SAM is the first model of its type in political science and provides three useful innovations over existing methods in computer science. First, we incorporate a ridge-like regularization that allows us to utilize many more features than permitted by existing approaches. Second, SAM is a hierarchical model that learns the flow between modes of speech, modeling each mode as a HMM and the transitions between these modes as a higher-level HMM. Third, we provide a principled approach to uncertainty by Bayesian bootstrap. We demonstrate SAM with three applications. First, we provide a benchmark for the model and demonstrate its generality by training SAM to classify segments according to who is speaking, where the mode of speech is the speaker ID. We then provide a second benchmark in which the labels are "speaker skepticism" instead of ID. Third, we classify justice speech during Supreme Court Oral Arguments according to whether or not the justice is using a skeptical tone, and show that this emotional content cannot be classified with only the text. We extend this analysis by examining the dynamics of emotional speech in Oral Arguments and suggest that rather than being strictly an automatic emotional response, speaker tone may be employed strategically within judicial bodies. We implement this model in an open-source R package, available upon publication.[1]

*We thank Dustin Tingley for research support through the NSF-REU program; Michael May, Thomas Scanlan, Angela Su, and Shiv Sunil for excellent research assistance; and the Harvard Experiments Working Group and the MIT Department of Political Science for generously contributing funding to this project. For helpful comments, we thank Justin de Benedictis-Kessner, Gary King, Connor Huff, In Song Kim, Dustin Tingley, and Teppei Yamamoto, as well as participants at the Harvard Applied Statistics Workshop and the International Methods Colloquium.

†Ph.D. Candidate, Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge MA 02139; dcknox.com, dcknox@mit.edu.

‡Ph.D. Candidate, Institute for Quantitative Social Science, 1737 Cambridge Street, Harvard University, Cambridge MA 02138; christopherlucas.org, clucas@fas.harvard.edu.

[1] Knox and Lucas (2017)

1 Introduction

Applications of text analysis in political science often examine corpora which were first spoken, then transcribed. To name but a few examples, in American and comparative politics, numerous articles study speech made by executives, legislators, and justices (Sigelman and Whissell, 2002*a*,*b*; Yu, Kaufmann and Diermeier, 2008; Monroe, Colaresi and Quinn, 2009; Quinn et al., 2010; Black et al., 2011; Proksch and Slapin, 2012; Eggers and Spirling, 2014; Kaufman, Kraft and Sen, ND). Though methodologically diverse, this research shares in common an exclusive focus on the words alone. However, human speech contains information beyond simply the spoken text. The rhetoric and tone of human speech conveys information that moderates the textual content (El Ayadi, Kamel and Karray, 2011*a*), and without appropriate methods to analyze the audio signal accompanying the text transcript, researchers risk overlooking important insights into the content of political speech. Moreover, studies of spoken speech span a range of fields, from speech made by elected officials (Proksch and Slapin, 2010) to deliberations and statements about foreign policy (Stewart and Zhukov, 2009; Schub, 2015).

Despite the frequency with which social scientists analyze speech, we are aware of no research in the social sciences that explicitly models the audio that accompanies these textual transcriptions. However, political scientists nonetheless study aspects of speech like emotion (Black et al., 2011) and rhetorical style (Sigelman and Whissell, 2002*a*,*b*), which depend on tone of speech as well as the words used (Scherer and Oshinsky, 1977; Murray and Arnott, 1993; Dellaert, Polzin and Waibel, 1996). And though methods for analyzing text as data have received a great deal of attention in political science in recent years (Laver, Benoit and Garry, 2003; Benoit, Laver and Mikhaylov, 2009; Clark and Lauderdale, 2010; Hopkins and King, 2010; Grimmer and Stewart, 2013; Lauderdale and Clark, 2014; Roberts et al., 2014; Lucas et al., 2015), none permit the inclusion of the accompanying audio features, even though recent work demonstrates the importance of audio features in political speech (Dietrich, Enos and Sen, 2016).

We fill this methodological void by proposing the speaker-affect model (SAM) for classifying the mode of speech connoted by the audio features that accompany text speech. SAM is a hierachical hidden Markov model, where modes of speech are modeled as a HMM, and transitions between these modes are modeled by a higher-level HMM. SAM is the first model

of its kind in political science and builds on approaches in computer science which use hidden Markov models for speech classification (Schuller, Rigoll and Lang, 2003; Nwe, Foo and De Silva, 2003; Nogueiras et al., 2001). We make three primary innovations over approaches in other fields. First, we incorporate a ridge-like regularization that allows us to utilize many more features than permitted by existing approaches, improving performance over models in computer science. Second, SAM is a hierarchical model that learns the flow between modes of speech, modeling each mode as a HMM and the transitions between these modes as a higher-level HMM. This modeling choice lets users analyze *speech dynamics*, the way rhetoric influences downstream speech, which is particularly interesting in cases of strategic interaction like those of interest to social scientists. Third, we provide a principled approach to uncertainty by Bayesian bootstrap, which is also of particular interest to social scientists interested in more than just predictive accuracy.

In addition to these statistical contributions, we make two empirical contributions, the first to literature in computer science on speech classification and the second to debates in political science about judicial speech. First, computer scientists typically validate their models with and analyze particularly "clean" data, typically recordings of actors paid to demonstrate various emotions, for example (Lee et al., 2004; Busso et al., 2004). However, for the model to be of use to social scientists, it must be successful with "real-world" data, which often implies much subtler labels. We demonstrate SAM with precisely such a task by classifying the speech of Supreme Court Justices according to the amount of skepticism in their tone of voice, showing that SAM is able to classify speech according to subtle, real-world categories. Second, Dietrich, Enos and Sen (2016) demonstrate that vocal pitch in questions by justices predict votes. In our application, we build upon this result in several ways. First, by modeling emotions with dozens of features rather than using a single feature (like vocal pitch) as a proxy, we're able to directly model the emotional categories of interest and validate classification of them. Second, we build on the predictive results in Dietrich, Enos and Sen (2016) by exploring speech dynamics; what are the effects of justice emotion and what strategic uses, if any, might such emotion have? We argue that justices use emotional rhetoric to position themselves on certain issues and to communicate those positions to their colleagues during the course of the case.

The remainder of this paper is as follows. In Section 2, we generally introduce audio as data to political science. Section 3 develops the model and inference, which is applied in Section 4. Finally, in Section 5, we conclude.

2 Audio as Data

In this section, we introduce audio as data to political science. As noted in Section 1, the number of papers developing and applying methods for text analysis has increased rapidly in recent years. However, little effort has been devoted to the analysis of other data signals that often accompany text. How can the accompanying audio be similarly treated "as data"? In this section, we describe the necessary steps, beginning with a description of raw audio, then explain how that signal is processed before it may be input into a model like SAM.

2.1 The Raw Audio Signal

The human speech signal is transmitted as compression waves through air. A microphone translates air pressure into an analog electrical signal, which is then converted to sequence of signed integers by pulse code modulation. This recording process involves sampling the analog signal at a fixed sampling rate and rounding to the nearest discrete value as determined by the audio bit depth, or the number of binary digits used to encode each sample value. Higher bit depths can represent more fine-grained variation.

In order to statistically analyze audio as data, we must first format and preprocess the recordings. Recordings are typically long and composed of multiple speakers. The model presented in this paper is developed for single-speaker segments, which can be computed by calculating time stamps for words in an associated transcript, if available. If the audio corpus of interest has not been transcribed, researchers can identify unique speakers with automated methods that rely on clustering algorithms to estimate the number of speakers and when they spoke in the recording. Single-speaker speech is then cut into sentence-length *utterances*, a segment of speech in which there are no silent regions. This further stage of segmentation is accomplished within the `R` package `SAM` (Knox and Lucas, 2017). For these speaker-utterances, we compute a series of *audio features*.

2.2 Raw Audio to Audio Features

We extract a wide range of features that have been used in the audio emotion-detection literature.[2] The raw audio signal is divided into overlapping 25-millisecond windows, spaced at 12.5-millisecond intervals. Some features, such as the sound intensity (measured in decibels) are extracted from the raw signal.

Next, features based on the audio frequency spectrum are extracted. The audio signal (assumed to be stationary within the short timespan of the window) is decomposed into components of various frequencies, and the power contributed by each component is estimated by discrete Fourier transform. The shape of the resulting power spectrum, particularly the location of its peaks, provides information about the shape of the speaker's vocal tract, e.g. tongue position. Some artifacts are introduced in this process, most notably by truncating the audio signal at the endpoints of the 25-millisecond frame and by the greater attenuation of high-frequency sounds as they travel through air. We ameliorate the former with a Hamming window that downweights audio samples toward the frame endpoints, and compensate for the latter using a pre-emphasis filter that boosts the higher-frequency components. Finally, we extract measures of voice quality, commonly used to diagnose pathological voice, based on the short-term consistency of pitch and intensity. Various interactions used in the emotion-detection literature are calculated, and the first and second finite differences of all features are also taken.

Table 1 shows the full set of features that we extract for each frame. As noted, we also include some interactions, as well as derivatives, which is possible because of the regularization step in SAM. The table divides features into those calculated directly from the raw audio, spectral features, and those measuring voice quality. Spectral features are those based on the frequency spectrum (for example, energy in the lower portion of the spectrum), while voice quality describes features that measure vocal qualities like "raspiness" and "airiness." Note as well that for some rows, we calculate many more than one feature. This is because the feature description describes a class of features, like energy in each of 12 pitch ranges, for example.

We group contiguous frames together into sentence-length *utterances*. When timestamped transcripts are available, as in our Supreme Court application in Section 4, we use them to

[2]For excellent reviews of the literature, including a more thorough discussion of these features, see Ververidis and Kotropoulos (2006); El Ayadi, Kamel and Karray (2011*b*).

Features from raw audio samples		
energy	1 feature / frame	sound intensity, in decibels: $\log_{10}\sqrt{\overline{x_i^2}}$
ZCR	1 feature / frame	zero-crossing rate of audio signal
TEO	1 feature / frame	Teager energy operator: $\log_{10}\overline{x_i^2 - x_{i-1}x_{i+1}}$
Spectral features		
F0	2 features / frame	fundamental, or lowest, frequency of speech signal (closely related to perceived pitch; tracked by two algorithms)
formants	6 features / frame	harmonic frequencies of speech signal, determined by shape of vocal tract (lowest three formants and their bandwidths)
MFCC	12 features / frame	Mel-frequency cepstral coefficients (based on discrete Fourier transform of audio signal, transformed and pooled to approximate human perception of sound intensity in 12 pitch ranges)
Voice quality		
jitter	2 features / frame	average absolute difference in F0
shimmer	2 features / frame	average absolute difference in energy

Table 1: Audio features extracted in each frame. In addition, we include interactions between (*i*) energy and zero-crossing rate, and (*ii*) Teager energy operator and fundamental frequency. We also use the first and second finite differences of all features.

segment the audio. Otherwise, speech can be segmented using a rule-based system to pick out brief pauses in continuous speech. Other classifiers can be trained to detect events of interest, such as interruptions or applause. We do so by coding a event-specific training set composed of the events of interest, as well as a few seconds before and after each instance to serve as a baseline. We then trained a linear support vector machine to classify individual audio frames as, for example, "applause" or "no applause." Framewise classifications are smoothed and thresholded to reduce false positives. This simple classifier is an effective and computationally efficient method for isolating short sounds with distinct audio profiles, such as an offstage voice. Continuous sections of speech by the same individual are thus isolated as separate segments. This allowed us to create single-speaker utterances for later analysis.

3 The Speaker-Affect Model

In this section, we introduce the speaker-affect model, or SAM. SAM is a hierarchical hidden Markov model (HHMM), meaning that each "state" in SAM is itself another hidden Markov model. Within SAM, states are the user-defined labels, like "angry" and "neutral" or "male"

and "female." Each of these states, by contrast, is modeled as an *unsupervised* HMM, learned during the training process. In the case of speech modes, this is useful because it permits each mode of speech to be defined by learned transitions between "sounds," which can be inferred from the user-supplied labels.

In the remainder of this section, we introduce our notation, define the model, and overview inference.

3.1 Notation

We assume a model of discrete speech modes, as is common in the emotion detection literature. However, in classifying political speech we depart from traditional models of so-called "basic" emotions such as anger or fear (Ekman, 1992, 1999), which are posited to be universal across cultures and often involuntarily expressed. Because such emotions are rare in political speech, of model of them is not especially useful. Instead, we argue that most actors of interest are professional communicators with a reasonable degree of practice and control over their speech. Political speakers generally employ more complex modes of speech, such as skepticism or sarcasm, in pursuit of context-specific goals such as persuasion or strategic signaling. To this end, we develop a method that can learn to distinguish between arbitrary modes of speech specified by subject-matter experts. This is a method for *speaker-dependent* emotion classification—that is, we do not assume that emotional signals are universal—based on prior instances of emotional speech by the same individual.

Our model segments continuous speech into utterances, generally bracketed by pauses. A speaker's mode of speech is assumed to be constant during an utterance. This is the quantity that we wish to measure, and it is generally unobserved unless a human coder listens to and classifies the utterance. Naturally, the mode of speech is not independent across utterances: A calm utterance is generally followed by another calm utterance. On a more granular level, each utterance is composed of an unobserved sequence of sounds, such as vowels, sibilants, and plosives. We. These sounds then generate a continuous stream of observed audio features.

Indices:

- Utterance index $u \in \{1, \cdots, U\}$: continuous segment of audible speech by a single speaker, preceded and followed by a period of silence or a transition between speakers.

- Time index $t \in \{1, \cdots, T_u\}$: position of audio window or video frame within an utterance. Advances by increments of 12.5 milliseconds.

Latent states:

- $S_u \in \{1, \cdots, M\}$: latent emotional state at for utterance u, corresponding to the emotions joy, sadness, anger, fear, surprise, disgust, and neutral. Indexed by m.
- $R_{u,t} \in \{1, \cdots, K\}$: latent sound/expression at time t (could correspond to, e.g., sibilant, plosive, grimace). Indexed by k. Note that the same index may take on different meanings depending on the emotional state. For example, sibilants may appear in both angry and neutral speech, but exact auditory characteristics will differ by emotion, and the index corresponding to the concept of "sibilant" may not be the same for each emotion.

Features:

- $\boldsymbol{X}_{u,t}$: column vector of D audio/visual features at time t, such as sound intensity (decibels) or position of mouth corners. All feature vectors in an utterance are collected in the $T_u \times D$ matrix, $\boldsymbol{X}_u$ (with $D = 189$ features in total: 27 audio features, 36 visual features, and first and second derivatives).

3.2 Model

We assume that the feature series is generated by a hierarchical hidden Markov model (HHMM) with two levels. First, a speaker's emotional state in utterance u is assumed to be drawn from a first-order HMM, i.e., randomly drawn based on the emotional state in the previous utterance. The probability of transitioning from emotion m to m' is given by $\Delta_{m,m'}$, and all transition probabilities are collected in the emotion transition matrix $\boldsymbol{\Delta}$.

$$S_u \sim \text{Cat}(\Delta_{\text{S}_{\text{u}-1},*})$$

Second, given that utterance u was spoken with emotion $S_u = m$, the sequence of sounds and expr(essions that comprise an utterance are assumed to be generated by the m-th emotion-specific first-order HMM. The probability of transitioning from sound/expression k to k' is given by $\Gamma^m_{k,k'}$, and transition probabilities are collected in sound/expression transition matrix

$\mathbf{\Gamma}^m$.

$$(R_{u,t} \mid S_u = m) \sim \text{Cat}(\Gamma^{\text{m}}_{\text{R}_{\text{u,t}-1},*})$$

Finally, during a particular sound/expression, the vector of features at each point in time is assumed to be drawn from a multivariate Gaussian distribution.

$$(X_{u,t} \mid S_u = m, R_{u,t} = k) \sim N(\boldsymbol{\mu}^{m,k}, \mathbf{\Sigma}^{m,k})$$

For example, the hypothetical "grimace" expression might have a low value for $\mu^{\text{anger, grimace}}_{\text{central lip thickness}}$, and the covariance matrix $\mathbf{\Sigma}^{\text{anger, grimace}}$ might contain a small variance for central lip thickness and positive covariance with left and right nostril heights. Given that the speaker is making an angry grimace, the exact facial pose on successive video frames are assumed to be independent draws from this distribution.[3]

We use superscripts to index the properties of states and sounds/expressions; subscripts index the elements of a vector or matrix.

3.3 Training

To estimate the parameters of the model, we select a training set of utterances for human coding. Each coder receives a subset of the training utterances, in random order, and classifies it into one of the M basic emotional states. Thus, during the training stage, the emotion labels, S_u, are known and the emotion-specific distributions are estimated.[4]

The training procedure consists of the following steps:

1. Estimate the parameters of the emotion-specific HMM distributions
   (a) Estimate the parameters of the sound/expression Gaussian distributions $\boldsymbol{\mu}^{m,k}$ and $\mathbf{\Sigma}^{m,k}$
   (b) Estimate the sound/expression transition matrix $\mathbf{\Gamma}^m$
2. Estimate the emotion transition matrix $\mathbf{\Delta}$

[3] Despite the seemingly implausible independence assumption, features may still exhibit substantial autocorrelation, because expressions tend to persist for multiple frames and different expressions often have very different mean values.

[4] In practice, because the perception of emotion is subjective, discrepancies between coders are to be expected and the emotion labels are not known precisely. We address this in the following section by weighting the contribution of an utterance to the model for emotion m by the proportion of human coders who classified the utterance as emotion m.

3. Select the optimal number of sounds/expressions in each emotion, K, via cross-validation

3.3.1 A Gentle Introduction: Training a Single HMM with a Single Utterance of Known Emotion

In this section, we first present a simplified version of our model—specifically, we start with a single utterance, and we assume that the emotion of that utterance is perfectly known. Some parameters in this section are marked with an overline to prevent confusion with the actual versions used in the following section, where we relax these constraints and present the actual procedure used in the training stage. The introduction to HMMs in this section is adapted from Zucchini and MacDonald (2009). Our full model is developed the following section.

Suppose that the speaker's emotional state during utterance u is known to be $S_u = m$. Audio and visual features, $\boldsymbol{X}_u = [X_{u,1}, \cdots, X_{u,T_u}]^\top$, are also observed. However, the underlying sounds and expressions that generated these features are completely unobserved: neither their labels ($R_{u,t}$, the order in which sounds and expressions occurred) nor their contents ($\boldsymbol{\mu}^{m,k}$ and $\boldsymbol{\Sigma}^{m,k}$, the auditory and visual characteristics of each sound/expression) are known. We treat the sound/expression labels as missing data and estimate by the expectation–maximization algorithm (EM).

For simplicity of exposition, we begin with a single utterance u, with known emotional state $S_u = m$. At each time t, the feature vector $\boldsymbol{X}_{u,t}$ could have been generated by any of the K sounds/expressions associated with emotion m, so there are K^{T_u} possible sequences of sounds/expressions by which the feature sequence, $\boldsymbol{X}_u$, could have been generated. The observed-data likelihood is the joint probability of all observed features is found by summing over every possible sequence of sounds and expressions:

$$\begin{aligned}
&\mathcal{L}^m(\boldsymbol{\mu}^{m,k}, \boldsymbol{\Sigma}^{m,k}, \boldsymbol{\Gamma}^m \mid \boldsymbol{X}_u, S_u = m) \\
&\quad = \Pr(\boldsymbol{X}_{u,1} = \boldsymbol{x}_{u,1}, \cdots, \boldsymbol{X}_{u,T_u} = \boldsymbol{x}_{u,T_u} \mid \boldsymbol{\mu}^{m,k}, \boldsymbol{\Sigma}^{m,k}, \boldsymbol{\Gamma}^m) \\
&\quad = \delta^{m\top} \boldsymbol{P}(\boldsymbol{x}_{u,1}) \left[\prod_{t=2}^{T_u} \boldsymbol{\Gamma}^m \boldsymbol{P}(\boldsymbol{x}_{u,t}) \right] \mathbf{1}, \qquad (1)
\end{aligned}$$

where δ^m is a $1 \times K$ vector containing the initial distribution of sounds/expressions (assumed to be the stationary distribution, a unit row eigenvector of $\boldsymbol{\Gamma}^m$), the matrices $\boldsymbol{P}(\boldsymbol{x}_{u,t}) \equiv$

$\text{diag}\left(\phi_D(\boldsymbol{x}_{u,t}; \boldsymbol{\mu}^{m,k}, \boldsymbol{\Sigma}^{m,k})\right)$ are $K \times K$ diagonal matrices in which the (k,k)-th element is the (D-variate Gaussian) probability of $x_{u,t}$ being generated by sound/expression k, and $\mathbf{1}$ is a column vector of ones. The parameters $\boldsymbol{\mu}^{m,k}$, $\boldsymbol{\Sigma}^{m,k}$, and $\boldsymbol{\Gamma}^m$ can in principle be found by directly maximizing this likelihood.

In practice, given the vast number of parameters to optimize over, we estimate using the Baum–Welch algorithm, a flavor of expectation–maximization for hidden Markov models. This procedure involves maximizing the complete-data likelihood, which differs from equation 1 in that it also incorporates the probability of the unobserved sounds/expressions.

$$
\begin{aligned}
&\Pr(\boldsymbol{X}_{u,1} = \boldsymbol{x}_{u,1}, \cdots, \boldsymbol{X}_{u,T_u} = \boldsymbol{x}_{u,T_u}, R_{u,1} = r_{u,1}, \cdots, R_{u,T_u} = r_{u,T_u} \mid \boldsymbol{\mu}^{m,*}, \boldsymbol{\Sigma}^{m,*}, \boldsymbol{\Gamma}^m) \\
&\quad = \delta_{r_{u,1}}^{m\top} \, \phi_D(\boldsymbol{x}_{u,1}; \boldsymbol{\mu}^{m,r_{u,1}}, \boldsymbol{\Sigma}^{m,r_{u,1}}) \times \\
&\qquad \prod_{t=2}^{T_u} \Pr(R_{u,t} = r_{u,t} \mid R_{u,t-1} = r_{u,t-1}) \, \phi_D(\boldsymbol{X}_{u,t}; \boldsymbol{\mu}^{m,r_{u,t}}, \boldsymbol{\Sigma}^{m,r_{u,t}}) \\
&\quad = \prod_{k=1}^{K} \left(\delta_k^{m\top} \, \phi_D(\boldsymbol{x}_{u,1}; \boldsymbol{\mu}^{m,k}, \boldsymbol{\Sigma}^{m,k})\right)^{\mathbf{1}\{R_{u,1}=k\}} \times \\
&\qquad \prod_{t=2}^{T_u} \left(\prod_{k=1}^{K} \left(\prod_{k'=1}^{K} \left(\Gamma_{k,k'}^m\right)^{\mathbf{1}\{R_{u,t}=k', R_{u,t-1}=k'\}} \, \phi_D(\boldsymbol{X}_{u,t}; \boldsymbol{\mu}^{m,k}, \boldsymbol{\Sigma}^{m,k})^{\mathbf{1}\{R_{u,t}=k\}} \right) \right),
\end{aligned} \tag{2}
$$

The algorithm uses the joint probability of (i) all feature vectors up until time t and (ii) the sound at t, given in equation 3. Together, these are referred to as the *forward probabilities*, because values for all t are efficiently calculated in a single recursive forward pass through the feature vectors.

$$
\begin{aligned}
\boldsymbol{\alpha}_{u,t} &\equiv \Pr(\boldsymbol{X}_{u,1} = \boldsymbol{x}_{u,1}, \cdots, \boldsymbol{X}_{u,t} = \boldsymbol{x}_{u,t}, R_{u,t} = k) \\
&= \delta_u^\top \boldsymbol{P}(\boldsymbol{x}_{u,1}) \left(\prod_{t'=2}^{t} \boldsymbol{\Gamma}^m \boldsymbol{P}(x_{u,t'}) \right)
\end{aligned} \tag{3}
$$

The algorithm also relies on the conditional probability of (i) all feature vectors after t given (ii) the sound/expression at t (equation 4). These are similarly called the *backward probabilities*

due to their calculation by backward recursion.

$$\begin{aligned}\boldsymbol{\beta}_{u,t} &\equiv \Pr(\boldsymbol{X}_{u,t+1} = \boldsymbol{x}_{u,t+1}, \cdots, \boldsymbol{X}_{u,T_u} = \boldsymbol{x}_{u,T_u} \mid R_{u,t} = k) \\ &= \left(\prod_{t'=t+1}^{T_u} \boldsymbol{\Gamma}^m \boldsymbol{P}(\boldsymbol{x}_{u,t'}) \right) \mathbf{1} \end{aligned} \tag{4}$$

3.3.2 E step

The E step involves substituting (i) the unobserved sound/expression labels, $\mathbf{1}\{R_{u,t} = k\}$, and (ii) the unobserved sound/expression transitions, $\mathbf{1}\{R_{u,t} = k\}$, with their respective expected values, conditional on the observed features $\boldsymbol{X}_u$ and the current estimates of $\boldsymbol{\mu}^{m,k}$, $\boldsymbol{\Sigma}^{m,k}$, and $\boldsymbol{\Gamma}^m$ (collectively referred to as $\boldsymbol{\Theta}$).

For (i), combining equations 1, 3 and 4 immediately yields the expected sound/expression label

$$\mathbb{E}\left[\mathbf{1}\{R_{u,t} = k\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}, S_u = m\right] = \tilde{\alpha}_{u,t,k} \; \tilde{\beta}_{u,t,k} \; / \; \tilde{\mathcal{L}}^m, \tag{5}$$

where the tilde denotes the current approximation based on parameters from the previous M step, and $\tilde{\alpha}_{u,t,k}$ and $\tilde{\beta}_{u,t,k}$ are the k-th elements of $\tilde{\boldsymbol{\alpha}}_{u,t}$ and $\tilde{\boldsymbol{\beta}}_{u,t}$, respectively.

For (ii), after some manipulation, the expected sound/expression transitions can be expressed as

$$\begin{aligned}&\mathbb{E}[\mathbf{1}\{R_{u,t} = k', R_{u,t-1} = k\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}, S_u = m] \\ &\quad = \Pr(R_{u,t} = k', R_{u,t-1} = k, \boldsymbol{X}_u \mid \tilde{\boldsymbol{\Theta}}) \; / \; \Pr(\boldsymbol{X}_u \mid \tilde{\boldsymbol{\Theta}}) \\ &\quad = \Pr(\boldsymbol{X}_{u,1}, \cdots, \boldsymbol{X}_{u,t-1}, R_{u,t-1} = k \mid \tilde{\boldsymbol{\Theta}}) \; \Pr(R_{u,t} = k' \mid R_{u,t-1} = k, \tilde{\boldsymbol{\Theta}}) \; \times \\ &\qquad \Pr(\boldsymbol{X}_{u,t} \mid R_{u,t} = k') \; \Pr(\boldsymbol{X}_{u,t+1}, \cdots, \boldsymbol{X}_{u,T_u} \mid R_{u,t} = k') \; / \; \Pr(\boldsymbol{X}_u \mid \tilde{\boldsymbol{\Theta}}) \\ &\quad = \tilde{\alpha}_{u,t-1,k} \; \tilde{\Gamma}^m_{k,k'} \; \phi_D(\boldsymbol{x}_{u,t}; \tilde{\boldsymbol{\mu}}^{m,k}, \tilde{\boldsymbol{\Sigma}}^{m,k}) \; \tilde{\beta}_{u,t,k'} \; / \; \tilde{\mathcal{L}}^m. \end{aligned} \tag{6}$$

3.3.3 M Step

After substituting equations 5 and 6 into the complete-data likelihood (equation 2), the M step involves two straightforward calculations.

First, the maximum likelihood update of the transition matrix $\boldsymbol{\Gamma}^m$ follows almost directly

from equation 6:

$$\left(\Gamma^m_{k,k'} \mid S_u = m\right) = \frac{\sum_{t=2}^{T_u} \mathbb{E}\left[\mathbf{1}\{R_{u,t} = k', R_{u,t-1} = k\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}, S_u = m\right]}{\sum_{t=2}^{T_u}\sum_{k'=1}^{K} \mathbb{E}\left[\mathbf{1}\{R_{u,t} = k', R_{u,t-1} = k\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}, S_u = m\right]} \tag{7}$$

Second, the optimal update of the k-th sound/expression distribution parameters found by fitting a Gaussian distribution to the feature vectors, with weights given by the expected value of the k-th label.

$$\left(\boldsymbol{\mu}^{m,k} \mid S_u = m\right) = \boldsymbol{X}_u^\top \overline{\boldsymbol{W}}_u^{m,k} \tag{8}$$

$$\left(\boldsymbol{\Sigma}^{m,k} \mid S_u = m\right) = \boldsymbol{X}_u^\top \mathrm{diag}\left(\overline{\boldsymbol{W}}_u^{m,k}\right) \boldsymbol{X}_u - \boldsymbol{\mu}^{m,k}\boldsymbol{\mu}^{m,k\top} \tag{9}$$

$$\text{where } \overline{\boldsymbol{W}}_u^{m,k} \equiv \frac{\left[\mathbb{E}\left[\mathbf{1}\{R_{u,1} = k\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}, S_u = m\right], \cdots, \mathbb{E}\left[\mathbf{1}\{R_{u,T_u} = k\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}, S_u = m\right]\right]^\top}{\sum_{t=1}^{T_u} \mathbb{E}\left[\mathbf{1}\{R_{u,t} = k\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}, S_u = m\right]}$$

3.3.4 Training Multiple Emotion-specific HMMs with Multiple Utterances of Imperfectly Observed Emotion

In practice, due to the subjective nature of perceived emotion, a speaker's true emotional state during a utterance is impossible to know with certainty. The emotional labels attached to training utterances inevitably contain inter-coder variation: some proportion of coders, $\hat{S}^{\text{train}}_{u,m}$, will classify an utterance as the m-th emotion. We treat $\hat{\boldsymbol{S}}^{\text{train}}_u = [\hat{S}^{\text{train}}_{u,1}, \cdots, \hat{S}^{\text{train}}_{u,M}]^\top$ as a probability vector over the speaker's emotional state during utterance u, which naturally leads to the use of $\hat{S}^{\text{train}}_{u,m}$ as a weighting factor in the estimation of the m-th emotion model. In the E step, we modify equation 5 to

$$\mathbb{E}\left[\mathbf{1}\{R_{u,t} = k, S_u = m\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}\right] = \hat{S}^{\text{train}}_{u,m}\ \tilde{\alpha}_{u,t,k}\ \tilde{\beta}_{u,t,k}\ /\ \tilde{\mathcal{L}}^m, \tag{10}$$

and equation 6 to

$$\begin{aligned}&\mathbb{E}[\mathbf{1}\{R_{u,t} = k', R_{u,t-1} = k, S_u = m\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}] \\ &\quad = \hat{S}^{\text{train}}_{u,m}\ \tilde{\alpha}_{u,t-1,k}\ \tilde{\Gamma}^m_{k,k'}\ \phi_D(\boldsymbol{x}_{u,t}; \tilde{\boldsymbol{\mu}}^{m,k}, \tilde{\boldsymbol{\Sigma}}^{m,k})\ \tilde{\beta}_{u,t,k'}\ /\ \tilde{\mathcal{L}}^m.\end{aligned} \tag{11}$$

The multi-utterance extensions of the M step equations 7, 8, and 9 are

$$\Gamma^m_{k,k'} = \frac{\sum_{u=1}^{U_{\text{train}}} \sum_{t=2}^{T_u} \mathbb{E}\left[\mathbf{1}\{R_{u,t} = k', R_{u,t-1} = k, S_u = m\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}\right]}{\sum_{u=1}^{U_{\text{train}}} \sum_{t=2}^{T_u} \sum_{k'=1}^{K} \mathbb{E}\left[\mathbf{1}\{R_{u,t} = k', R_{u,t-1} = k, S_u = m\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}\right]} \tag{12}$$

$$\boldsymbol{\mu}^{m,k} = \sum_{u=1}^{U_{\text{train}}} \boldsymbol{X}_u^\top \boldsymbol{W}_u^{m,k} \tag{13}$$

$$\boldsymbol{\Sigma}^{m,k} = \sum_{u=1}^{U_{\text{train}}} \left(\boldsymbol{X}_u^\top \text{diag}\left(\boldsymbol{W}_u^{m,k}\right) \boldsymbol{X}_u\right) - \boldsymbol{\mu}^{m,k} \boldsymbol{\mu}^{m,k\top} \tag{14}$$

$$\text{where } \boldsymbol{W}_u^{m,k} \equiv \frac{\sum_{u=1}^{U_{\text{train}}} \left[\mathbb{E}\left[\mathbf{1}\{R_{u,1} = k, S_u = m\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}\right], \cdots, \mathbb{E}\left[\mathbf{1}\{R_{u,T_u} = k, S_u = m\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}\right]\right]^\top}{\sum_{u=1}^{U_{\text{train}}} \sum_{t=1}^{T_u} \mathbb{E}\left[\mathbf{1}\{R_{u,t} = k, S_u = m\} \mid \boldsymbol{X}_u, \tilde{\boldsymbol{\Theta}}\right]}$$

3.4 Extrapolation

In section 3.3.4, we discussed a procedure to learn how a speaker generates audio/visual features while speaking with a particular emotion. This procedure is based on a training set of utterances with human-coded emotions, and its output is M separate, emotion-specific hidden Markov models, each with some estimated parameters $\hat{\boldsymbol{\mu}}^{m,k}$, $\hat{\boldsymbol{\Sigma}}^{m,k}$, and $\hat{\boldsymbol{\Gamma}}^m$. In this section, we show how emotion-specific HMMs can be used to predict the emotions of utterances in previously unseen speech.

This task can be approached in two ways: we can either (i) "decode" the most likely *local* emotion of the u-th utterance, or (ii) decode the most likely *sequence* of emotions that generated all U utterances *globally*. In our applications, the two approaches produce results that differ in insubstantial ways, if at all.

In either case, the general procedure to estimate the higher-level HMM broadly parallels the preceding sections. The main difference is that emotional labels are available for the training set, which eliminates the need for the E step. Instead of assuming a the stationary distribution, we simply estimate the frequency of each emotion, $\overline{\boldsymbol{S}} \equiv \frac{1}{U_{\text{train}}} \sum_{u=1}^{U_{\text{train}}} \hat{\boldsymbol{S}}_u^{\text{train}}$, using the human-coded training utterances. Using consecutive training utterances, we also estimate the emotional transition matrix $\hat{\boldsymbol{\Delta}}$, by

$$\hat{\Delta}_{m,m'} = \frac{\sum_{u=1}^{(U_{\text{train}}-1)} j \hat{S}^{\text{train}}_{u,m} \; \hat{S}^{\text{train}}_{u+1,m'} \; \mathbf{1}\{u \text{ immediately precedes } u+1\}}{\sum_{u=1}^{(U_{\text{train}}-1)} \hat{S}^{\text{train}}_{u,m} \mathbf{1}\{u \text{ immediately precedes } \; u+1\}} \tag{15}$$

As before, we calculate the that the feature sequence in utterance u, was generated by the

estimated model for the m-th emotion: $\Pr\left(\boldsymbol{X}_u \mid \hat{\boldsymbol{\Theta}}, S_u = m\right)$. These state probabilities are collected in the diagonal matrix $\boldsymbol{P}(\boldsymbol{X}_u)$.

As before, we define the total, forward, and backward probabilities,

$$\begin{aligned}
\mathcal{L} &= \Pr(\boldsymbol{X}_1 = \boldsymbol{x}_1, \cdots, \boldsymbol{X}_U = \boldsymbol{x}_U) \\
&= \overline{\boldsymbol{S}}^\top \boldsymbol{P}(\boldsymbol{x}_1) \left(\prod_{u'=2}^{U} \hat{\boldsymbol{\Delta}} \boldsymbol{P}(\boldsymbol{x}_{u'}) \right) \mathbf{1} \qquad (16) \\
\boldsymbol{A}_u &= \Pr(\boldsymbol{X}_1 = \boldsymbol{x}_1, \cdots, \boldsymbol{X}_u = \boldsymbol{x}_u, S_u = m) \\
&= \overline{\boldsymbol{S}}^\top \boldsymbol{P}(\boldsymbol{x}_1) \left(\prod_{u'=2}^{u} \hat{\boldsymbol{\Delta}} \boldsymbol{P}(\boldsymbol{x}_{u'}) \right) \qquad (17) \\
\boldsymbol{B}_u &= \Pr(\boldsymbol{X}_{u+1} = \boldsymbol{x}_{u+1}, \cdots, \boldsymbol{X}_U = \boldsymbol{x}_U \mid S_u = m) \\
&= \left(\prod_{u'=u+1}^{U} \hat{\boldsymbol{\Delta}} \boldsymbol{P}(\boldsymbol{x}_{u'}) \right) \mathbf{1} \qquad (18)
\end{aligned}$$

The local probabilities of the emotional states are then

$$\left[\Pr\left(S_u = m \mid \boldsymbol{X}, \hat{\boldsymbol{\Theta}}\right)\right] = \boldsymbol{A}_u \boldsymbol{B}_u / \mathcal{L} \qquad (19)$$

and the globally most likely sequence of emotional states is found by the Viterbi algorithm, a dynamic programming approach that efficiently finds the sequence of local emotions that maximizes the probability of the observed features.

3.4.1 Assorted Complications

The above glosses over some practical issues that arise in the estimation of our model.

Numerical Issues:

HMMs are particularly vulnerable to numerical underflow in the calculation of forward, backward, and total probabilities; for a discussion, see Zucchini and MacDonald (2009).

Because we work with a large number of features, the M step involves fitting a high-dimensional Gaussian distribution for each sound/expression. When a sound/expression is relatively rare, this can occasionally lead to sample covariance matrices that are non-invertible. We address this by ridge-like regularization, or adding a small value to the diagonal of the sample covariance; the value of this regularization parameter is determined by cross-validation.

Like Gaussian mixture models, HMMs are sometimes subject to the "collapse" of smaller components. For example, a sound/expression distribution may move directly over a single data point, and its variance may go to zero; this results in a contribution to the likelihood that is infinite for one data point and zero for all others. We address this problem by detecting collapse by setting a tolerance parameter for the determinant of the covariance matrix (roughly the "volume" of the state distribution) and randomly resetting the state to a randomly chosen position with a large, spherical covariance matrix. This requires standardizing all features before estimation, so that "volume" and the value of the tolerance parameter can be defined in a meaningful way.

Convergence:

The likelihood functions of most HMMs can suffer from degeneracy, and the EM algorithm is not guaranteed to converge to a global maximum. We run multiple EM chains for all estimates, select results from the chain that converges to the highest likelihood, and use the variation between chains as a diagnostic.

Missing data:

Features are frequently subject to partial missingness. For example, the fundamental frequency is notoriously difficult to estimate under certain conditions. Different tracking algorithms often produce widely varying results, and gender-based heuristics are generally needed to rule out implausible estimates. When this occurs, we treat the fundamental frequency as missing data; however, other features may still be available for the affected frames. Similarly, visual features may be obscured when a speaker turns away from the camera. We assume that missingness is at random, conditional on the remaining observed features. To calculate probabilities for these partially missing frames, we integrate over all possible values that the missing data could have taken on.

Model selection:

To select the regularization parameter and the number of sounds/expressions in an emotion, we divide the training utterances into five folds. We then employ V-fold cross-validation and choose the number of sounds/expressions that optimize the total likelihood of the validation sets (van der Laan et al., 2004). This procedure possesses the "oracle" property in that it asymptotically selects the closest approximation, in terms of the Kullback–Leibler divergence,

to the true data-generating process among the candidate models considered.

4 Applications

In this section, we demonstrate the application of `SAM` with two applications, both of which use an original corpus of audio scraped from the Oyez Project.[5] In the first, we benchmark SAM against the only alternative approach for audio classification presently available in R or Python (`pyAudioAnalysis`) (Giannakopoulos, 2015). In this analysis, we classify utterances of speech according to the identity of the speaker. In the second application, we classify utterances according to their emotional characteristics, and again show that against `pyAudioAnalysis`, SAM performs considerably better.

4.1 Data

The data for these applications are scraped from the Oyez Project.[6] We limit our analysis to the Roberts court from the Kagan appointment to the death of Justice Scalia, so that the same justices are on the court for the entirety of the period we analyze. The Oyez data contains an accompanying text transcript, as well as time stamps for utterance start and stop times and speaker labels. We use these timestamps to segment the audio into utterances in which there is a single speaker. However, occasionally, segment stop times are earlier than the stop times, due to errors in the data provided by Oyez. In these sections, we drop the full section of speech in which this speaker was speaking. To validate the remaining segments, we employ two procedures. First, we randomly sample 100 segments from the remaining segments and manually listen to them. In these cases, 100% of the sampled segments were correct. Second, we manually code 1,100 randomly sampled segments (100 per Justice, after removing Justice Thomas). When doing so, we include an option for "incorrect segment." Among these 1,100 segments, we found 0 incorrect segments. Given both of these procedures, there is little reason to doubt the resulting segments.

[5] Dietrich, Enos and Sen (2016) separately and concurrently scraped the same audio data and conducted a novel analysis of it, which we build on here.

[6] https://www.oyez.org/

4.2 Application 1: Speaker Identification Benchmark

In our first application, we validate the effectiveness of SAM in a relatively simple audio classification task, speaker identification. Our general approach is to learn 11 separate models, one for each justice's overall speech, based on a set of labeled training utterances. We then evaluate a new set of test utterances to determine the justice model that is most likely to have generated each test utterance. The speaker is then predicted based on this most likely model. In order to evaluate performance through cross-validation, we randomly assign utterances into separate folds. This process of randomization breaks the original ordering of utterances. As a result, these predictions are based only the lower level HMMs and does not incorporate higher-level information about the flow of speech between speakers in a conversation. We test SAM against a number of alternative audio models in the pyAudioAnalysis (Giannakopoulos, 2015) library. We choose pyAudioAnalysis because it is currently the only alternative audio classifier available in R or Python.

The general approach taken by pyAudioAnalysis differs from SAM in that it does not attempt to model the dynamics of speech. Instead, pyAudioAnalysis summarizes each utterance as a single feature vector, where each element represents a particular summary statistic for the entire utterance. Results show that SAM performs significantly better across all possible parameter settings: even the worst-performing (simplest and most heavily regularized) SAM performs markedly better than the best of 24 pyAudioAnalysis models. This is true not only in overall accuracy, but also in the recall rate for any individual speaker.

We visualize the results of this benchmark in two ways. First, in Figure 1, we plot the predictive accuracy in speaker ID classification of all the models we run. The top panel (red dots) displays the results of SAM across all model parameters the we run. The remaining panels correspond to alternate models implemented in pyAudioAnalysis (support vector machines, extremely randomized trees, random forest, and gradient boosting), again across parameter values. The key insight from this experiment is that *the worst predictions from SAM outperformed the best predictions across all other models.*

Second, we break this result down by label (i.e., speaker names) over two confusion tables, shown in Figures 2a and 2b. Figure 2a shows the confusion table for the best model from pyAudioAnalysis (a SVM with C = 0.01), while Figure 2b shows the best from SAM. These

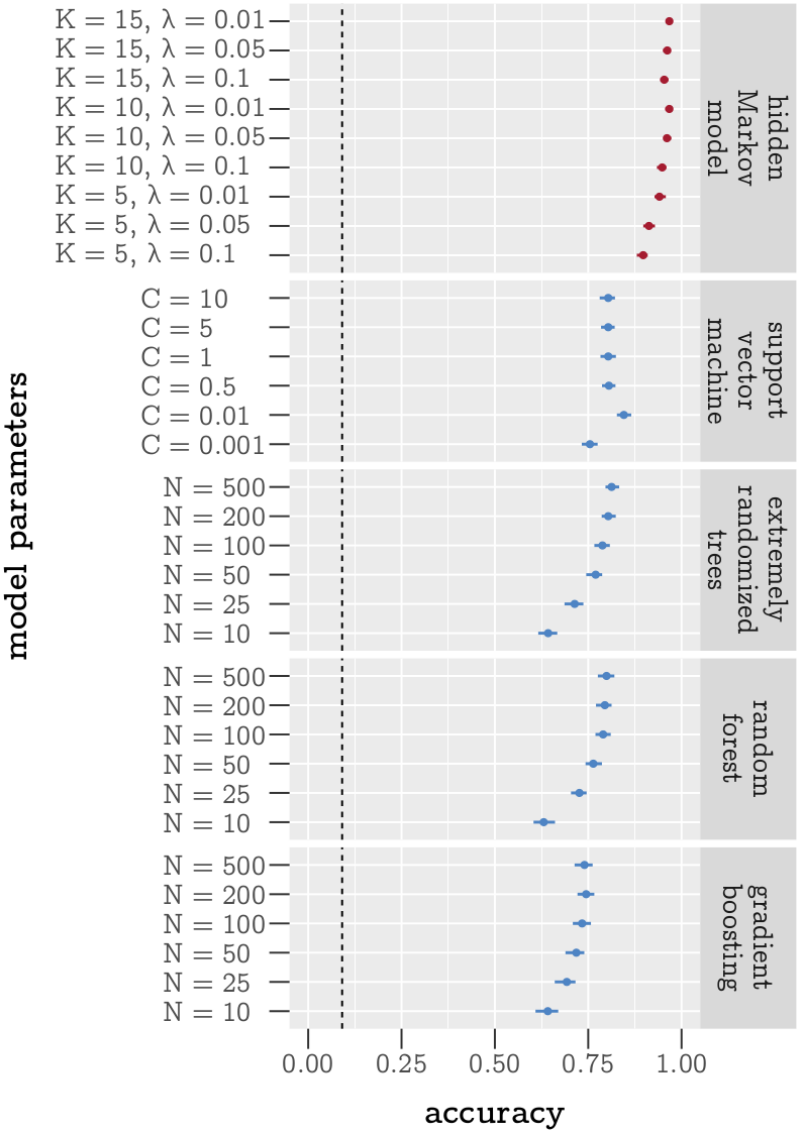
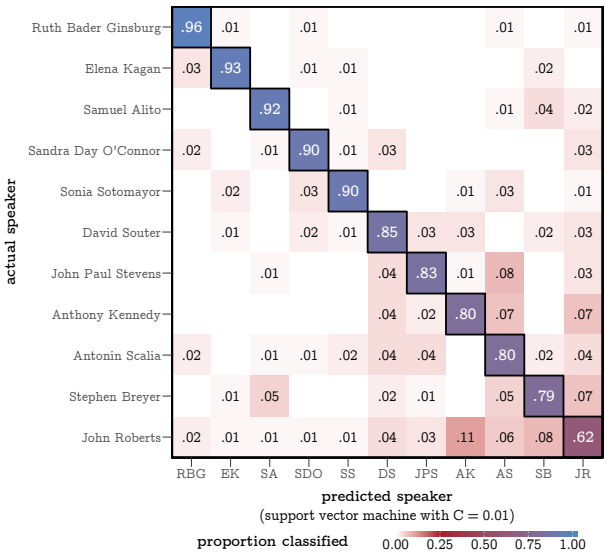
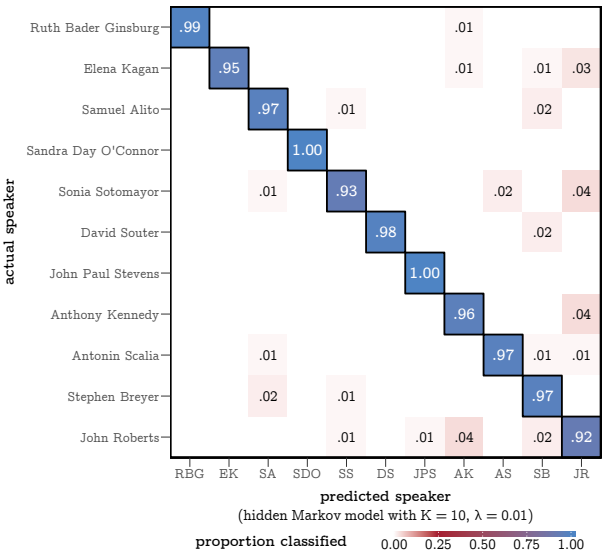


Figure 1: Predictive accuracy of SAM (red) versus all models available in pyAudioAnalysis, across various parameter values. Note that the worst-performing model from SAM outperforms the best from pyAudioAnalysis.



(a) Confusion table for the best classifier from pyAudioAnalysis, a SVM with C = 0.01.



(b) Confusion table for the best classifier from SAM, with K = 10 and $\lambda = 0.01$.

plots provide several key insights. First, note that SAM outperforms the best pyAudioAnalysis model on each speaker. Even though the SVM does a good job with Justices Ginsburg and Kagan, for instance, SAM does even better. Second, pyAudioAnalysis performs very poorly on some speakers, most notably Justice Roberts, successfully classifying his segments only 62% of the time. By contrast, SAM successfully classifies Roberts 92% of the time, and performs even better on the rest of the justices.

4.3 Application 2: Emotion Recognition Benchmark

Next, we conduct a second benchmark experiment, demonstrating that SAM can correctly distinguish between different modes of speech by the same speaker. In the context of Supreme Court speech, this is a considerably harder tasks and is a direct validation of the substantive analysis that we undertake in the Subsection 4.4. Specifically, we train SAM to classify instances "skeptical" speech.

Skepticism is a particularly interesting rhetorical category. As Johnson, Wahlbeck and Spriggs (2006, p.99) argue, justices use oral arguments to "seek information in much the same way as members of Congress, who take advantage of information provided by interest groups and experts during committee hearings to determine their policy options or to address uncertainty over the ramifications of making a particular decision." With these intentions in

mind, recent work analyzes how justice pitch when asking questions during oral arguments Dietrich, Enos and Sen (2016) and the text of those questions Kaufman, Kraft and Sen (ND) predict that justice's vote on the respective case. We build on these results by providing the first direct classifier of a particular rhetorical mode, namely skepticism. Skepticism is especially interesting if, as Johnson, Wahlbeck and Spriggs (2006) argue, justices use oral arguments to seek information, because skepticism is a subtle yet direct measure of the concepts and arguments that justices are willing to doubt (Taber and Lodge, 2006), which is theoretically distinct from more neutral-toned questions, in that the latter does not imply an oppositional view on the topic, whereas a question asked in a skeptical tone implies to the lawyer and the other justices that the issue at hand is not believable. Ability to measure skeptical tone, then, introduces to the literature on courts and decision-making in judicial bodies a method the permits the study of questions about when and why justices *doubt* arguments made in the courtroom, rather than simply when and why they ask questions.

We create our training set according to the following rules. For each speaker, we listen to 100 utterances. Then, for each utterance, we ask if the tone of voice suggests that the justice is skeptical, ignoring the text of the question. For example, in the landmark case Obergefell v. Hodges, Justice Scalia asks, "And how many States have – have voted to have same-sex marriage or their legislature or – or by referendum? I think it's 11, isn't it?" From the text, it is not clear if he is skeptical or not. However, the utterance occurs amidst a back and forth with the Mary Bonauto, who was representing the petitioners, in which Justice Scalia is expressing skepticism at the idea that gay marriage should be interpreted as a constitutional requirement rather than an issue to be decided by the states. Though this skepticism is unclear from the text, it is quite clear in his tone of voice, and we code this passage accordingly.

After constructing the training set, we train a series of models with SAM and select the SAM model with the highest out-of-sample F1 score. The F1 score is a widely used measure of classification performance that incorporates both precision and recall by taking their harmonic mean. As a result, this procedure tends to select models that both correctly label truly skeptical speech as such and do not raise many false alarms. The optimal SAM model, as selected through V-fold cross-validation, achieved an overall accuracy of 70%. Moreover, the performance of this model was well-balanced across different modes of speech: 71% (70%) of

truly skeptical (neutral) speech was labeled correctly.

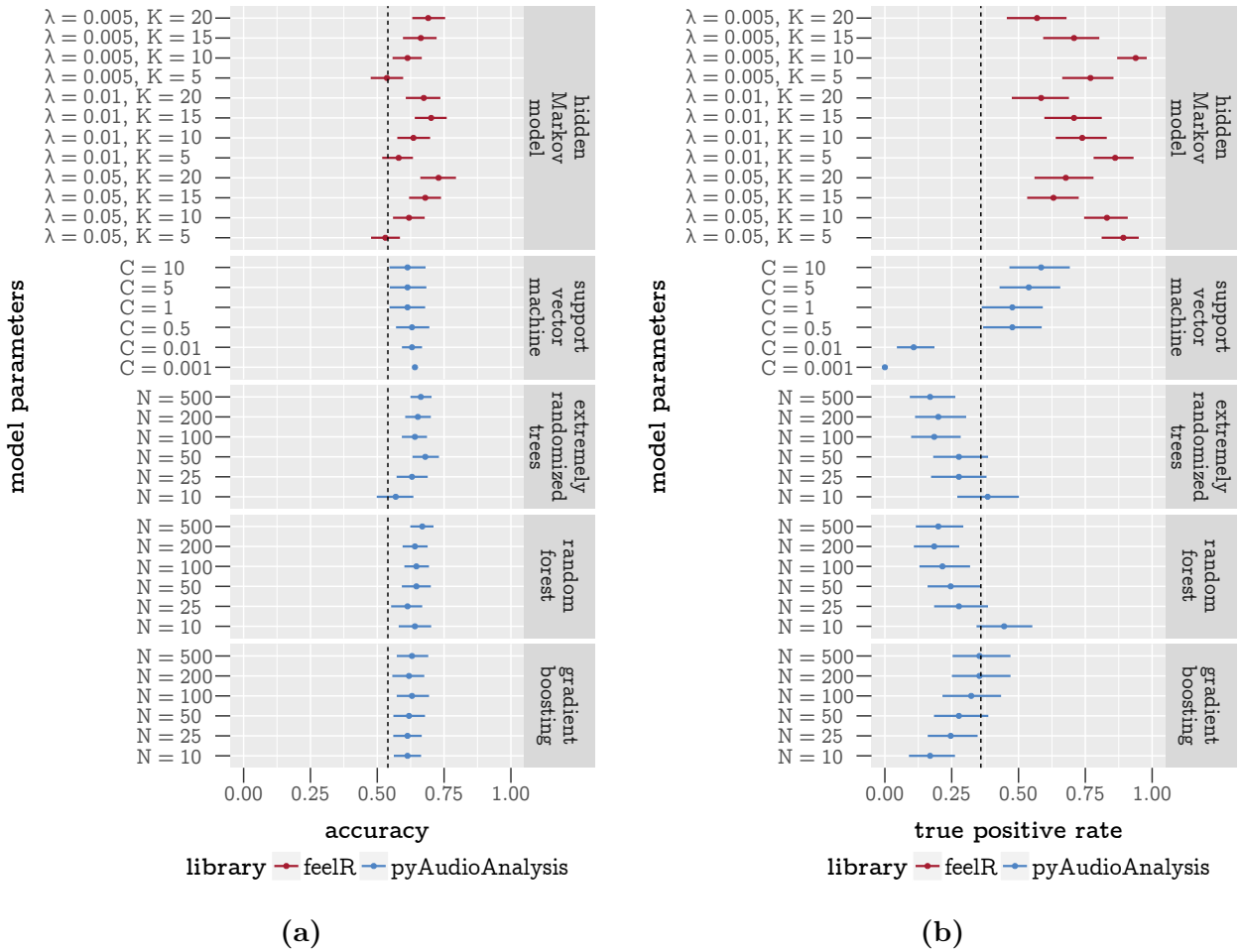
We again compare SAM's performance against pyAudioAnalysis. We find, as in Subsection 4.2, that the optimal pyAudioAnalysis model performs worse in overall accuracy (61%). However, the most notable differences lie in the models' true positive rates. Because Roberts' neutral utterances are roughly twice as frequent, these models tended to fall back on a "neutral" guess when they had difficulty distinguishing between the modes of speech. This strategy performs reasonably well from the perspective of overall accuracy, but poorly in terms of correctly detecting instances of the less-prevalent mode.

Because of the subtlety of a category like skeptical speech, classifiers that, unlike SAM, do not directly model speech dynamics fair poorly. To illustrate how difficult this task is, we compare models from pyAudioAnalysis against randomly guessing according to overall emotion prevalance (i.e., guessing the common categories), and find that random guessing common categories actually outperforms most models in pyAudioAnalysis. This is why some models in pyAudioAnalysis can achieve high accuracy but low true positive rates. For example, one SVM model achieved a 64% overall accuracy rate by simply guessing "neutral" every time, although the true positive rate of this model was naturally zero as a result.

Again, we plot performance of SAM against models from pyAudioAnalysis, shown in Figures 3a and 3b. Figure 3a shows the results of model accuracy. The important insight from these plots is that classifers like SVM are able to achieve high accuracy by guessing the common category (in this case, neutral). This can be seen in Figure 3b. In both plots, the dotted vertical line shows accuracy of guessing. While pyAudioAnalysis succeeds in overall accuracy, many of these models perform worse than guessing. By contrast, all SAM estimates are considerably better than guessing. Given that instances of skeptical speech are precisely the quantity of interest, this difference is noteworthy. This point is further illustrated in Figure 4, which plots the true positive rate on the predictive accuracy. For tasks like classifying emotion in speech, it is clear that SAM outperforms alternative models.

4.3.1 Comparison with Text Sentiment

Given the amount of research on text and the courts, we also compare SAM to text-based sentiment analysis using the corresponding transcripts provided by Oyez. However, 100 utter-



(a) **(b)**

Figure 3: Performance of SAM (red) against pyAudioAnalysis (blue). The left panel displays overall model accuracy, while the right shows the true positive rate. Models from pyAudioAnalysis succeed in overall accuracy by guessing the common category, while SAM is able to achieve comparable overall accuracy while significantly outperforming models from pyAudioAnalysis in its ability to correctly classify rare categories (in this case, skeptical speech). Given that instances of skeptical speech are precisely the quantity of interest, this difference is noteworthy.

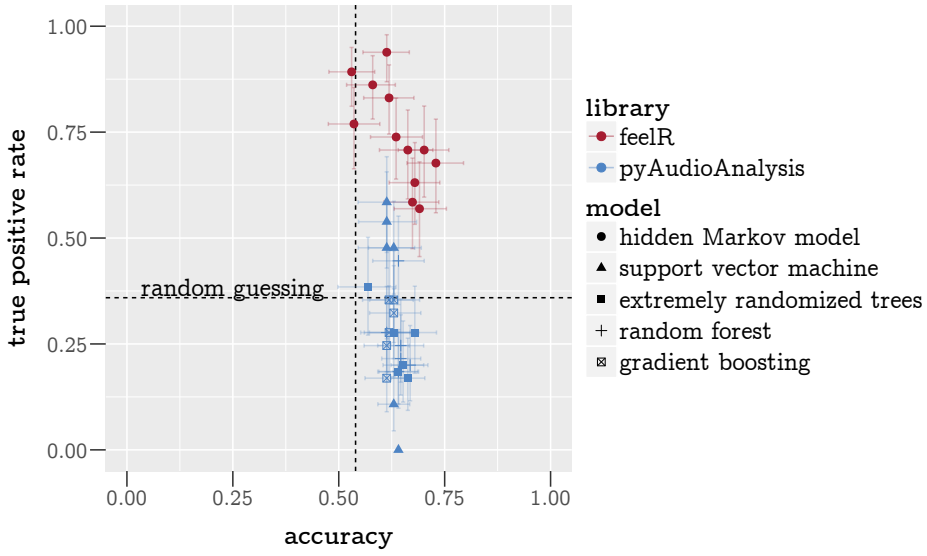


Figure 4

ances per speaker is suffciently small that it is effectively impossible to train an even remotely plausible text classifier. For example, we attempted to train an SVM on our hand-coded utterances (the same training set used in the preceeding audio benchmarks) but were unable to get even remotely plausible results. This is another argument in favor of using the audio data, as it can in fact be more informative in small samples for classification tasks like ours.

Given that we cannot effectively train a text classifier, we consider instead using a pretrained sentiment classifier. Specificlaly, we use a state-of-the-art deep learning model, the recursive neural network (Socher et al., 2011), in which a treebank is employed to represent sentences based on their structures. Because the data in this case are too few to train our own Recursive Neural Network, we use pretrained weights provided Socher et al. (2013). Based on the the transcribed text, the neural network generates one of five possible labels for each utterance: "very negative", "negative", "neutral", "positive", and "very positive". We pool the two negative categories and treat these as predicting skepticism, because this produces the most favorable possible results for the neural network. Using this classification scheme, 78% of utterances are classified as skeptical, which leads to overall accuracy of 45% (much lower than all audio classifiers), a true positive rate of 89% (higher, because nearly all utterances were positively classified), and a true negative rate of 20% (again, much lower, because few utterances were classified negatively).

[cross-validation for other justices in progress]

4.4 Application 3: Main Analysis

[in progress]

5 Conclusion

In this paper, we introduced a new hierarchical hidden Markov model, the speaker-affect model, for classifying modes of speech using audio data. With novel data of Supreme Court oral arguments, we demonstrated that SAM consistently outperforms alternate methods of audio classification, and further showed that especially when training data are small, text classifiers are not a viable alternative for identifying modes of speech. The approach we develop has a broad range of possible substantive applications, from speech in parliamentary debates (Goplerud, Knox and Lucas, 2016) to television news reporting on different political topics. With other interesting results on the importance of audio as data (Dietrich, Enos and Sen, 2016) accumulating, our approach is a useful and general solution that improves on existing approaches and broadens the set of questions open to social scientists.

References

Benoit, Kenneth, Michael Laver and Slava Mikhaylov. 2009. "Treating words as data with error: Uncertainty in text statements of policy positions." *American Journal of Political Science* 53(2):495–513.

Black, Ryan C, Sarah A Treul, Timothy R Johnson and Jerry Goldman. 2011. "Emotions, oral arguments, and Supreme Court decision making." *The Journal of Politics* 73(2):572–581.

Busso, Carlos, Zhigang Deng, Serdar Yildirim, Murtaza Bulut, Chul Min Lee, Abe Kazemzadeh, Sungbok Lee, Ulrich Neumann and Shrikanth Narayanan. 2004. Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th international conference on Multimodal interfaces*. ACM pp. 205–211.

Clark, Tom S and Benjamin Lauderdale. 2010. "Locating Supreme Court opinions in doctrine space." *American Journal of Political Science* 54(4):871–890.

Dellaert, Frank, Thomas Polzin and Alex Waibel. 1996. Recognizing emotion in speech. In *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*. Vol. 3 IEEE pp. 1970–1973.

Dietrich, Bryce J, Ryan D Enos and Maya Sen. 2016. Emotional Arousal Predicts Voting on the US Supreme Court. Technical report Technical Report.

Eggers, Andrew C and Arthur Spirling. 2014. "Ministerial Responsiveness in Westminster Systems: Institutional Choices and House of Commons Debate, 1832–1915." *American Journal of Political Science* 58(4):873–887.

Ekman, Paul. 1992. "An argument for basic emotions." *Cognition and Emotion* 6:169–200.

Ekman, Paul. 1999. Basic Emotions. In *Handbook of Cognition and Emotion*, ed. T. Dalgleish and M. Power. Chicester, England: Wiley.

El Ayadi, Moataz, Mohamed S Kamel and Fakhri Karray. 2011*a*. "Survey on speech emotion recognition: Features, classification schemes, and databases." *Pattern Recognition* 44(3):572–587.

El Ayadi, Moataz, Mohamed S. Kamel and Fakhri Karray. 2011*b*. "Survey on speech emotion recognition: features, classification schemes, and databases." *Pattern Recognition* 44:572–587.

Giannakopoulos, Theodoros. 2015. "pyaudioanalysis: An open-source python library for audio signal analysis." *PloS one* 10(12):e0144610.

Goplerud, Max, Dean Knox and Christopher Lucas. 2016. "The Rhetoric of Parliamentary Debate." *Working Paper* .

Grimmer, Justin and Brandon M Stewart. 2013. "Text as data: The promise and pitfalls of automatic content analysis methods for political texts." *Political Analysis* .

Hopkins, Daniel J and Gary King. 2010. "A method of automated nonparametric content analysis for social science." *American Journal of Political Science* 54(1):229–247.

Johnson, Timothy R, Paul J Wahlbeck and James F Spriggs. 2006. "The influence of oral arguments on the US Supreme Court." *American Political Science Review* 100(01):99–113.

Kaufman, Aaron, Peter Kraft and Maya Sen. ND. "Machine Learning and Supreme Court Forecasting: Improving on Existing Approaches.".

Knox, Dean and Christopher Lucas. 2017. "SAM: R Package for Estimating Emotion in Audio and Video." *Working Paper* .

Lauderdale, Benjamin E and Tom S Clark. 2014. "Scaling politically meaningful dimensions using texts and votes." *American Journal of Political Science* 58(3):754–771.

Laver, Michael, Kenneth Benoit and John Garry. 2003. "Extracting policy positions from political texts using words as data." *American Political Science Review* 97(02):311–331.

Lee, Chul Min, Serdar Yildirim, Murtaza Bulut, Abe Kazemzadeh, Carlos Busso, Zhigang Deng, Sungbok Lee and Shrikanth Narayanan. 2004. Emotion recognition based on phoneme classes. In *Interspeech*. pp. 205–211.

Lucas, Christopher, Richard A Nielsen, Margaret E Roberts, Brandon M Stewart, Alex Storer and Dustin Tingley. 2015. "Computer-Assisted Text Analysis for Comparative Politics." *Political Analysis* .

Monroe, Burt L, Michael P Colaresi and Kevin M Quinn. 2009. "Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict." *Political Analysis* p. mpn018.

Murray, Iain R and John L Arnott. 1993. "Toward the simulation of emotion in synthetic speech: A review of the literature on human vocal emotion." *The Journal of the Acoustical Society of America* 93(2):1097–1108.

Nogueiras, Albino, Asunción Moreno, Antonio Bonafonte and José B Mariño. 2001. Speech emotion recognition using hidden Markov models. In *INTERSPEECH*. pp. 2679–2682.

Nwe, Tin Lay, Say Wei Foo and Liyanage C De Silva. 2003. "Speech emotion recognition using hidden Markov models." *Speech communication* 41(4):603–623.

Proksch, Sven-Oliver and Jonathan B Slapin. 2010. "Position taking in European Parliament speeches." *British Journal of Political Science* 40(03):587–611.

Proksch, Sven-Oliver and Jonathan B Slapin. 2012. "Institutional foundations of legislative speech." *American Journal of Political Science* 56(3):520–537.

Quinn, Kevin M, Burt L Monroe, Michael Colaresi, Michael H Crespin and Dragomir R Radev. 2010. "How to analyze political attention with minimal assumptions and costs." *American Journal of Political Science* 54(1):209–228.

Roberts, Margaret E, Brandon M Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson and David G Rand. 2014. "Structural Topic Models for Open-Ended Survey Responses." *American Journal of Political Science* 58(4):1064–1082.

Scherer, Klaus R and James S Oshinsky. 1977. "Cue utilization in emotion attribution from auditory stimuli." *Motivation and emotion* 1(4):331–346.

Schub, Robert. 2015. Are You Certain? Leaders, Overprecision, and War. Technical report Working Paper (available at http://scholar. harvard. edu/schub/research).

Schuller, Björn, Gerhard Rigoll and Manfred Lang. 2003. Hidden Markov model-based speech emotion recognition. In *Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on*. Vol. 1 IEEE pp. I–401.

Sigelman, Lee and Cynthia Whissell. 2002*a*. "" The Great Communicator" and" The Great Talker" on the Radio: Projecting Presidential Personas." *Presidential Studies Quarterly* pp. 137–146.

Sigelman, Lee and Cynthia Whissell. 2002*b*. "Projecting presidential personas on the radio: An addendum on the Bushes." *Presidential Studies Quarterly* 32(3):572–576.

Socher, Richard, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, Christopher Potts et al. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. Vol. 1631 Citeseer p. 1642.

Socher, Richard, Cliff C Lin, Chris Manning and Andrew Y Ng. 2011. Parsing natural scenes and natural language with recursive neural networks. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. pp. 129–136.

Stewart, Brandon M and Yuri M Zhukov. 2009. "Use of force and civil–military relations in Russia: an automated content analysis." *Small Wars & Insurgencies* 20(2):319–343.

Taber, Charles S and Milton Lodge. 2006. "Motivated skepticism in the evaluation of political beliefs." *American Journal of Political Science* 50(3):755–769.

van der Laan, Mark J, Sandrine Dudoit, Sunduz Keles et al. 2004. "Asymptotic optimality of likelihood-based cross-validation." *Statistical Applications in Genetics and Molecular Biology* 3(1):1036.

Ververidis, imitrios and Constantine Kotropoulos. 2006. "Emotional speech recognition: Resources, features, and methods." *Speech Communication* 48:1162–1181.

Yu, Bei, Stefan Kaufmann and Daniel Diermeier. 2008. "Classifying party affiliation from political speech." *Journal of Information Technology & Politics* 5(1):33–48.

Zucchini, Walter and Iain MacDonald. 2009. *Hidden Markov Models for Time Series*. Boca Raton, FL: CRC Press.