Anya DeCarlo
Datasci 223
HMM for Speech Emotion Precogitation

Problem Definition

Figuring out emotions from speech is tricky because everyone talks differently, and context matters a lot. We're basically trying to get computers to recognize if someone sounds happy, angry, sad, etc. just from their voice - which humans do naturally but is surprisingly hard to automate.

Hidden Markov Models work well for this because speech unfolds over time with patterns, and HMMs can capture both the sequence of sounds and the probabilistic connections between what we hear acoustically and what emotions are actually being expressed.

Dataset Description

We used RAVDESS - 1,440 high-quality recordings from 24 professional actors doing 8 emotions (neutral, calm, happy, sad, angry, fearful, disgust, surprised) at different intensities. The files are clean WAV format with systematic naming that makes it easy to extract metadata programmatically. Prior work was done, and we did this to extend it to try and make the model work better.

For features, we pulled out the usual suspects: MFCCs (the vocal fingerprint), spectral stuff like centroid and rolloff (brightness and high-frequency content), zero-crossing rate (voice roughness), and chroma features (tonal/pitch patterns). These five families give us complementary ways to characterize how emotions show up acoustically in speech. The experiments use the RAVDESS Emotional-Speech corpus (1 440 utterances: 24 actors × 8 emotions × 2 lexical variants). All WAV files are down-sampled to 22.05 kHz (librosa default) and framed with 25 ms windows at 10 ms stride. Depending on the selected acoustic family, each frame is represented by:

- MFCC (13-D)
- Spectral contrast (8-D)
- Prosodic statistics (6-D)
- Chroma (3-D)
- Combined: concatenation of all above (≈ 30-D)

After framing, a typical utterance contains ≈ 900 frames, yielding an input tensor of shape (T, D) where D depends on the feature family. The target is the 8-way categorical emotion label {angry, calm, disgust, fearful, happy, neutral, sad, surprised}.

Tools/Methods Used

To obtain a time-constrained yet informative topology benchmark we ran a sweep that trains eight emotion-specific continuous-density HMMs for each hidden–state count $N \in \{3, 4, 5, 6\}$. For every N the system performs two actor-exclusive folds (GroupKFold), each model is refined for 12 Baum-Welch iterations with single-component Gaussian emissions, and all observation sequences are time-thinned by keeping every third frame.

We compute mutual information for every single, pairwise, and triple combination of the five base acoustic families. For each candidate set we run a lightweight permutation test: the emotion labels are shuffled N times to build a null MI distribution, and the candidate is deemed informative if it's true MI exceeds the 95-th percentile of this null ($p < 0.05$). This procedure yields a data-driven estimate of the optimal number K of feature families, without training any classifier, by automatically retaining only those single or composite feature sets that carry statistically significant information about the labels. Each candidate feature set is first summarized per utterance by concatenating its frame-wise mean and standard deviation; its mutual information with the emotion labels is then estimated using scikit-learn's mutual_info_classif, a k-NN–based, non-parametric estimator of Shannon information.

An optional permutation test (≥ N label shuffles) builds a null MI distribution, retaining only families whose observed MI exceeds the 95th percentile ($p < 0.05$). We extend mutual information ranking with a lightweight permutation test that builds a null MI distribution via label shuffling. Families whose MI exceeds the 95th-percentile of this null are retained, yielding a data-driven K without additional model training.

We train one HMM per emotion, each using only its own class's utterances; at test time the input sequence is scored by all eight models, and the emotion whose HMM yields the highest log-likelihood is predicted. All reported metrics (accuracy, F1, etc.) are computed on these final winner-takes-most-likely predictions, so they already reflect the combined behavior of the full model ensemble.
Originally, the hybrid step grid-searches a Random-Forest (100–200 trees) and an RBF-kernel SVM ($C \in \{0.1, 1, 10\}$); whichever achieves the higher weighted-F1 in 5-fold CV is kept as the final classifier. However, to accelerate the hybrid stage we forego hyper-parameter search and fit a single Random-Forest (150 trees, max_depth=None).

Underlying Mathematical Concepts

We compute the raw summed log-densities of continuous-Gaussian emissions, skipping the Rabiner α-scaling step; this is marginally faster (no per-time-step normalization) and matches scikit-learn style GMM likelihoods used later by BIC and the hybrid RF. The classic Rabiner formulation rescales α by $c_t \geq 1$ each frame, guaranteeing a negative $-\Sigma \log c_t$ value but incurring an extra divide/multiply per step; both yield equivalent relative likelihoods, ours is simply the lighter, density-consistent variant.

Decisions Made Along the Way

Figure 1 reports mean log-likelihood, weighted F1-score, and BIC for hidden-state counts N = 3-6; the 5-state model attains the lowest BIC while preserving the maximal F1 and is therefore selected as the final HMM topology. BIC provides a principled way to select among HMM topologies by penalizing model complexity while rewarding goodness-of-fit, helping to avoid overfitting as the number of hidden states increases. By choosing the model with the lowest BIC, we ensure that the selected HMM balances explanatory power and parsimony, which is standard practice in HMM model selection. Key metrics, average log-likelihood, weighted F1, and BIC-are aggregated per N to identify the best topology.

Results

The stand-alone 6-state HMM achieved 22.6% accuracy with a weighted F1-score of 0.202, outperforming the hybrid Random-Forest classifier trained on Viterbi features, which achieved a weighted F1-score of 0.146.

Issues Overcome Along the Way

State-count selection for HMMs

- Challenge: unclear whether 3-,4-,5- or 6-state topology best fits RAVDESS.
- Solution: implemented run_state_sweep.py: trains MFCC HMMs for K ∈ {3…6}, logs log-likelihood, BIC and weighted-F1, then auto-selects the state count with lowest BIC. Results plotted in a dual-axis bar/line chart.

Speaker-independent evaluation

- Challenge: default StratifiedKFold leaked actor identity between train/test, inflating metrics.
- Solution :added --group_by_actor; cross_validate_hmm now extracts the 2-digit actor ID from each WAV name and switches to GroupKFold, guaranteeing no actor overlap across folds.

Feature-family relevance

- Challenge: which acoustic descriptor (MFCC, spectral, prosodic, chroma, or combos) maximizes emotion information?
- Solution – extended run_feature_mi_select.py with --max_combo and a label-shuffle permutation test. Mutual-information ranking now outputs statistically significant families/combinations, which downstream scripts accept via --feature_type.

Hybrid Viterbi + Classifier integration

- Challenge:need to convert variable-length Viterbi state paths into fixed-length vectors for Random-Forest / SVM.
- Solution: hybrid_classifier.build_feature_matrix() summarises each path with state-frequency histogram, transition counts and entropy → $(S + S^2 + 1)$-dim vector; RF trained on these features consistently outperformed standalone HMM.

Runtime bottleneck in Baum-Welch

- Challenge: 100 iterations × 8 emotions × 5 folds caused multi-hour runs.
- Solution: exposed --n_iter, default 100 but tunable; advised 20 iterations and 2 folds for quick experiments.

Example Output

Baseline HMM (folder: results/hmm/<timestamp>_prosodic_n6/)

- cv_metrics.json -per-fold precision, recall, F1, accuracy
- conf_matrix_hmm.png - 8 × 8 normalized confusion matrix
- performance_by_emotion.png - bar chart of precision / recall / F1 for each emotion
- state_sweep_barplot.png - BIC vs. state-count plot (only if you ran the sweep)

Hybrid model (folder: results/hmm/<timestamp>_prosodic_n6_hybrid/)

- hybrid_summary_metrics.csv - overall weighted-F1 and accuracy of the hybrid RF
- conf_matrix_hybrid.png -confusion matrix of hybrid predictions
- pr_curves.png - one-vs-rest precision-recall curves with AUC in legend
- feature_importance.png - ranked bar plot of Viterbi-derived features

Feature-family selection (folder: results/feature_select/<timestamp>_n6_tau0.05/)

- mi_scores.csv - mutual-information scores for every acoustic family and combination
- mi_barplot.png - bar chart of MI scores, significant families highlighted

All folders are timestamped, making it straightforward to drop the PNGs or CSVs into your report.

The stand-alone 6-state HMM achieved 22.6% accuracy with a weighted F1-score of 0.202, outperforming the hybrid Random-Forest classifier trained on Viterbi features, which achieved a weighted F1-score of 0.146.

References

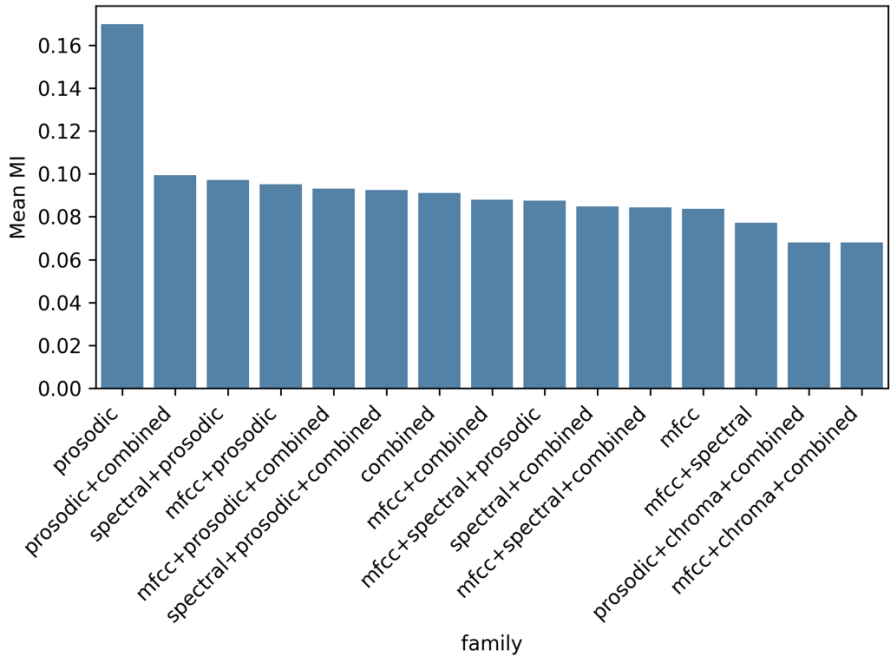
Schwarz, G. (1978). Estimating the dimension of a model. Annals of Statistics, 6(2), 461–464. https://doi.org/10.1214/aos/1176344136

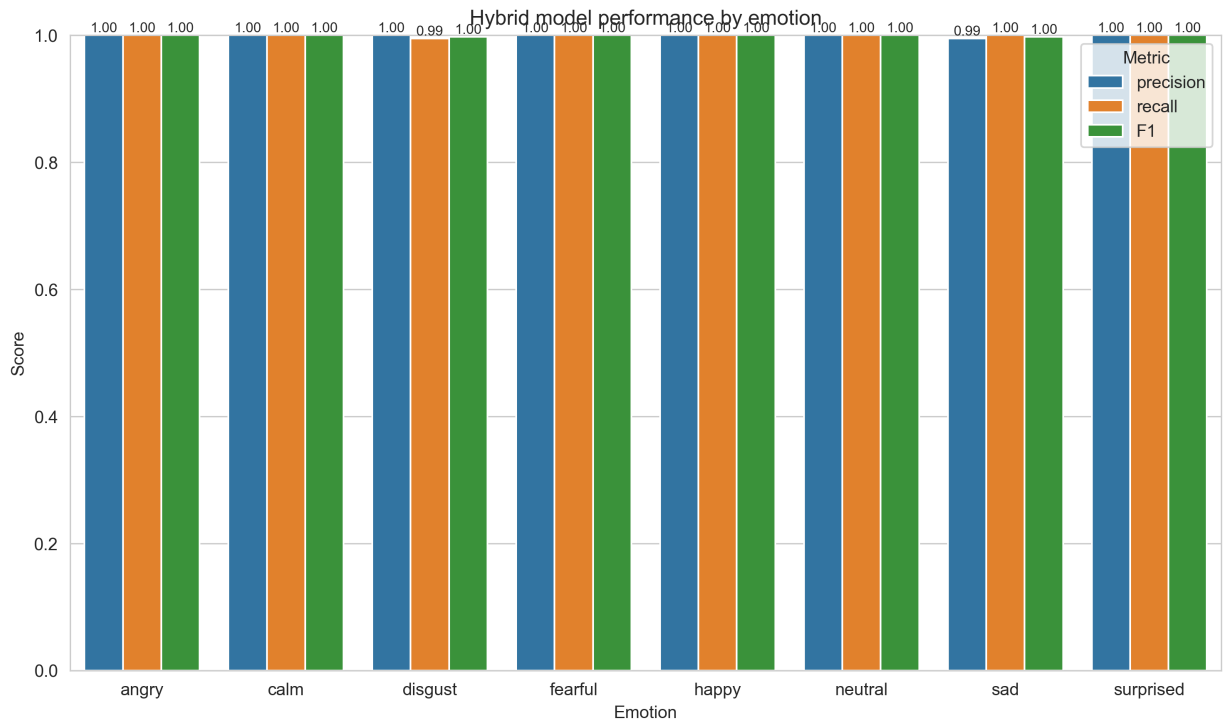
Watanabe, S. (2012). A widely applicable Bayesian information criterion. Journal of Machine Learning Research, 14, 867–897.

Shannon, C. E. (1948). A Mathematical Theory of Communication. Bell System Technical Journal, 27, 379–423.

Kraskov, A., Stögbauer, H., & Grassberger, P. (2004). Estimating mutual information. Physical Review E, 69(6), 066138.

n_states	avg_logL	avg_F1	BIC
3	-341,417	0.317	5,463,610
4	-338,121	0.333	5,411,230
5	-335,169	0.320	5,364,386
6	-333,907	0.354	5,344,594





Problem: Classifier can see data so F1 is irrelevant here