SPEAKER RECOGNITION SYSTEM

CS 280 Mini Project

Roy Amante Salvador
Isabelle Tingzon

December 3, 2015

Department of Computer Science

INTRODUCTION

Speaker recognition is the process of automatically extracting, characterizing, and identifying a speaker's identity based on the information available in his or her speech signal.

The human voice contains acoustic patterns and characteristics that differ from person to person.

- anatomical structure of the vocal tract
- speaking style and accents
- frequency or pitch of voice

These distinctive features can be used to identify a speaker.

APPLICATION OF SPEAKER RECOGNITION

Automatic speech recognition systems allow for users to verify their identity which has potential value in a range of applications.

- **Security and identity management.** Voice activated commands, voice biometrics for access control and authentication.
- **E-commerce.** Telephone banking, customer recognition.
- **Law Enforcement and Criminal Investigation.** e.g. comparing the voice of an assailant against a database of suspects to find the closest match
-

PURPOSE OF THE STUDY

The goal of this project is to implement an **Automatic Speaker Recognition System** using features:

- Mel Frequency Cepstral Coefficients (MFCC)
- and Spectral Subband Centroids (SSC)

and apply an array of classification algorithms including:

- Binary SVM
- Multi-class SVM
- Decision Trees
- Naive Bayes classifier

We will then compare their performances in terms of speed and accuracy.

METHODOLOGY

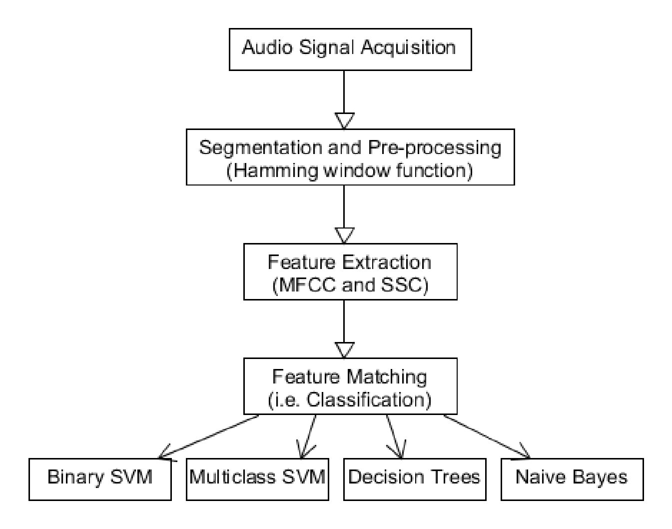

Figure: Overview of the System

Nine (9) users were asked to record their voices via the microphone and the system interface. Each user was asked to pronounce two sets of words (for training and testing) with a varied range of vowel sounds.

Table: Training Set

A	E	I	O	U
bag	egg	piece	on	use
cat	set	fit	pawn	book
par	mend	bin	ton	root
fan	fence	int	law	hoop
ant	ebb	reek	ore	use

Table: Test Set

A	E	I	O	U
sad	end	ill	boss	hue
pal	zen	mint	ode	soot
cab	sense	feel	mode	cute
wham	well	pick	lawn	swoon
arg	lent	wig	ohm	loom

The audio signal recording is sampled at 44100 Hz using stereo channels and saved in signed 16 bit wave file format.

We divide the audio signal into several overlapping frames. We used 10 ms overlap or skip size between 30 ms frames. Each frame is applied with a **Hamming window function H(N)**.

Hamming window is defined as

$$H(t) = 0.54 - 0.46 \cos\left(\frac{2\pi t}{N-1}\right) \quad 0 \leq t \leq N-1$$

where N is the frame size. Our signal then becomes

$$w(N) = x(N)H(N)$$

where x(N) is the raw audio signal of the frame.

Since we only expect the audio signal to contain the voice of the speaker, we employ a voice activity threshold θ to determine which frames the user is speaking. We define voice activity in the frame α as

$$\alpha = \frac{\sum_{\mathrm{i}=1}^{\mathrm{N}} |\mathrm{w}(\mathrm{i})|}{\mathrm{N}}$$

If the windowed frame doesn't meet the threshold i.e. $\alpha \leq \theta$, **it is considered a silent frame** and is ignored. We find $\theta = 200$ to be a suitable threshold value.

FEATURE EXTRACTION

For frames meeting the voice activity threshold described in the previous section, we extract the following features:

- Mel Frequency Cepstral Coefficients (MFCC)
- Spectral Subband Centroids (SSC)

using an available open source library Python Speech Features.

We used its default number of features - 13 for MFCC and 26 for SSC.

In this paper, we use several classification algorithms including

- Binary SVM
- Mutli-class SVM
- Decisions Trees
- Naive Bayes Classifier

for the feature matching phase of our speaker recognition system.

Support vector machines (SVM) was our first choice for classification since they are known to be among the most robust of classification algorithms

1. **Binary SVM** - for n classes, we require one SVM classifier per speaker/class k such that each classifier is trained on examples with label +1 if class is k; 0 otherwise. In testing, the SVM with the highest rank is the 'winner'. (LibSVM)
2. **Multi-class SVM** - Python toolkit scikit-learn

- **Decision Trees (CART)** - Decision trees are constructed using features and threshold that yield the largest information gain at each node. scikit-learn uses an optimized version of the CART algorithm.
- **Gaussian Naive Bayes classifier** - computes the likelihood of features using the following formula

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{\left(-\frac{(x_i-\mu_y)}{2\pi\sigma_y^2}\right)}$$

where μ_y and σ_y are estimated using maximum likelihood. (scikit-learn)

RESULTS AND DISCUSSION

The following metrics were used:

- **Rank.** - primary metric is the rank.
  Obtaining a rank 1 means the system is able to correctly identify the speaker.
- **Score / Accuracy.** - secondary metric
  This is defined as the ratio of the number of correctly identified frames for the speaker and the number of the total frames meeting the voice activity threshold of an audio clip.

Note: If the score of the speaker is the highest among all the other speakers in the database despite having a low score, it is still considered a correct classification.

FEATURES RESULTS

Training and Test Set Performance using Mel Frequency Cepstral Coefficients (MFCC) and Spectral Subband Centroids (SSC) features

	MFCC		SSC		MFCC + SSC	
	Rank	Score	Rank	Score	Rank	Score
Training Set						
Speaker 1	1	89.93%	1	99.71%	1	100%
Speaker 2	1	94.33%	1	99.89%	1	100%
Speaker 3	1	90.34%	1	99.78%	1	100%
Speaker 4	1	93.00%	1	99.79%	1	100%
Speaker 5	1	91.20%	1	99.60%	1	100%
Speaker 6*	1	85.59%	1	99.46%	1	100%
Speaker 7	1	92.14%	1	99.79%	1	100%
Average	**1**	**90.93%**	**1**	**99.72%**	**1**	**100%**
Test Set						
Speaker 1	1	70.34%	1	67.84%	1	73.09%
Speaker 2	1	68.44%	1	64.29%	1	62.54%
Speaker 3	1	61.44%	1	60.36%	1	62.32%
Speaker 4	1	66.31%	1	59.59%	1	67.01%
Speaker 5	1	79.68%	1	68.96%	1	73.49%
Speaker 6*	5	24.26%	1**	51.31%	1**	43.87%
Speaker 7	1	84.03%	1	74.36%	1	79.96%
Average	**1.57**	**64.93%**	**1**	**63.82%**	**1**	**66.04%**

* had noisy background

** near miss. Rank 2 within 5% distance

CLASSIFIER RESULTS

TABLE IV. TRAINING TIME AND PERFORMANCE OF DIFFERENT CLASSIFIERS USING MEL FREQUENCY CEPSTRAL COEFFICIENTS (MFCC) WITH SPECTRAL SUBBAND CENTROIDS (SSC) FEATURES

	Binary SVM per Speaker		One vs Rest Multiclass SVM		CART Decision Tree		Naive Bayes Classifier	
Training Time	2-24 seconds per SVM		12 seconds		4 seconds		$<$ 1 second	
	Rank	Score	Rank	Score	Rank	Score	Rank	Score
Training Set								
Speaker 1	1	100%	1	51.93%	1	100%	1	41.74%
Speaker 2	1	100%	1	46.56%	1	100%	1	71.02%
Speaker 3	1	100%	1	55.57%	1	100%	1	49.78%
Speaker 4	1	100%	1	36.08%	1	100%	1	42.13%
Speaker 5	1	100%	1	53.30%	1	100%	1	26.56%
Speaker 6*	1	100%	1	59.72%	1	100%	1**	29.83%
Speaker 7	1	100%	1	51.01%	1	100%	1	38.49%
Speaker 8	1	100%	1	86.84%	1	100%	1	72.75%
Speaker 9	1	100%	1	83.82%	1	100%	1	91.73%
Average	**1**	**100%**	**1**	**58.31%**	**1**	**100%**	**1**	**51.56%**

CLASSIFIER RESULTS [CONT'D]

TABLE IV. TRAINING TIME AND PERFORMANCE OF DIFFERENT CLASSIFIERS USING MEL FREQUENCY CEPSTRAL COEFFICIENTS (MFCC) WITH SPECTRAL SUBBAND CENTROIDS (SSC) FEATURES

Training Time	Binary SVM per Speaker 2-24 seconds per SVM		One vs Rest Multiclass SVM 12 seconds		CART Decision Tree 4 seconds		Naive Bayes Classifier $<$ 1 second	
	Rank	Score	Rank	Score	Rank	Score	Rank	Score
Test Set								
Speaker 1	1	77.31%	1	52.38%	1	43.63%	1	40.69%
Speaker 2	1	68.71%	1	48.65%	1	35.75%	1	75.79%
Speaker 3	1	68.48%	1	40.55%	1	34.63%	1	33.82%
Speaker 4	1	70.45%	1	33.48%	1	31.52%	1	40.58%
Speaker 5	1	76.81%	1	51.67%	1	45.27%	1**	22.93%
Speaker 6*	1**	49.14%	1**	31.44%	1**	26.20%	3	7.06%
Speaker 7	1	82.16%	1	60.21%	1	45.71%	1	37.66%
Speaker 8	1	90.13%	1	75.55%	1	76.51%	1	60.66%
Speaker 9	1	95.33%	1	80.59%	1	75.23%	1	92.77%
Average	**1**	**75.39%**	**1**	**52.72%**	**1**	**46.05%**	**1.22**	**45.77%**

* had noisy background

** near miss. Rank 2 within 5% distance

DISCUSSION OF RESULTS

For the training set, MFCC scored high at around 90% average and the SSC feature scored very high at 99%. **Combination of the two allowed the classifier yielded the best results.**

Speaker 6 anomaly:

- In the test set, MFCC generally outperforms SSC in terms of score with the exception of Speaker 6 who has audible background noise in his training audio clip.
- MFCC ranked and scored poorly with Speaker 6 but SSC was able to rank the Speaker at number 1.

MFCC with SCC outperformed both individual MFCC and SSC in terms of average score and was able to rank all the Speakers at number 1.

Generally, all classifiers performed well in ranking.

Although our Binary SVM took the longest time to train, it outperformed the other three classifiers significantly in terms of average score.

Only Naive Bayes Classifier missed recognizing speaker 6 which is the one with a lot of background noise.

The other three correctly classified all the speakers in both training and test set.

CONCLUSION

This paper presents a **successful implementation of an automatic speaker recognition system** using features MFCC and SCC trained on binary SVM, multi-class SVM, decision trees, and Naive Bayes classifier.

Overall, **all the classifiers worked well**, ranking the correct speaker as number 1 for all test samples **except for the Naive Bayes classifier** which misclassifies speaker 6.

Moreover, our findings show that the

- **Binary SVM** achieved the highest average score (or accuracy) of 75.39%.
- Combination of **MFCC and SSC yields the best performance** compared to using MFCC or SSC alone.

FUTURE WORK

Future work includes using an **audio noise reduction system** for better quality audio recordings.

In the real world setting, capturing background noises is inevitable; thus noise reduction methods will help reduce instances of misclassification (such that in the case of speaker 6) and will lead to an overall better classification performance.

END OF PRESENTATION