Gender Recognition by Voice and Speech Analysis

Jingyu Shao (shaojy15@ucla.edu)
UID: 204723890
Department of Statistics
University of California, Los Angeles
Los Angeles, CA 90095 USA

Abstract

The physical cues underlying the perception of speech can be used for various purposes. This paper aims at identifying a voice as male or female, based upon acoustic properties of the voice and speech. We designed several feature engineering approaches to first create corresponding prototypes for each category. Then we fed them into several classical supervised learning classification models, together with self-designed recurrent neural networks, to train a powerful classifier for gender recognition, and also conducted large-scale experiments to compare performances between different computational models and humans.

Keywords: speech perception; gender recognition; supervised learning; recurrent neural network

Introduction

Discrepancies between male and female voices are related to implicit and multidisciplinary issues, including not only acoustic measurements (mean frequency or magnitude), but also sociology or psychology (behavioral differences across genders). Our work focuses on the acoustic differences because they are straight-forward and sufficient for ordinary gender recognition problems.

Mean fundamental frequency is commonly considered to be the main discrepancy between male and female voice. The voiced speech of a typical adult male will have a fundamental frequency from 85 to 180 Hz, and that of a typical adult female from 165 to 255 Hz. But this may slightly vary through different ages, and even has relations with personal lifestyle. For example, those with a smoking habit have broadly lower voice frequency.

Also, the acoustic discrepancy across genders varies when it comes to different languages. For instance, (Johnson, 2008) reveals that the across-gender variations in acoustic properties are relatively small in Danish but seem to be much more obvious in Russian. The difference even varies when it comes to different dialects. In a Chinese dialect, mean F_0 is almost equivalent for male and female speakers (Rose, 1991); statistical analysis also reveals that Min speakers had a significantly greater maximum range of speaking intensity and a smaller lowest speaking intensity than Mandarin speakers (Chen, 2005).

In our study, we focus on selecting the most appropriate computational model for voice gender discrimination in challenging circumstances. We also aim at finding suitable feature engineering approaches for feeding the statistical models.

Related Work

Gender Recognition from Speech

Inferring gender from speech is necessary and has potential benefits for many down-stream applications. For example, the gender information can be used as part of a sex-dependent speech recognition system. Also, it serves as an important feature that can help greatly reduce the search space when doing speaker identification.

People have been working on gender recognition for decades (Childers, Wu, Bae, & Hicks, 1988; Zeng, Wu, Falk, & Chan, 2006; Meena, Subramaniam, & Gomathy, 2013). In (Zeng et al., 2006), a Gaussian mixture model is proposed and shown to be robust to noise, as well as independent of language. In (Meena et al., 2013), a neural network model is used. Different from our work, they used a multi-layer perceptron with one hidden layer, while we proposed a new model based on recurrent neural networks that can handle variable-length input. We also provide a comprehensive analysis of various feature engineering techniques and machine learning models.

Binary Classification

Classification is perhaps the most common problem in machine learning. The goal of classification is to assign a label y to each input data x. For obvious reasons there are only two labels in gender recognition, in which case we can let the label space to be $\{0,1\}$.

Various models can be used to do binary classification. They can either be discriminative, which model the conditional probability $p(y|x)$, or generative, which model the joint probability $p(y,x)$. In our work we will experiment with models from both classes.

Feature Engineering

As a typical binary-classification problem, we need to first create statistical features from inexpressible data formats such as videos and audios with relative domain knowledge. We proposed 2 main approaches for audio data preprocessing.

Dataset Preparation

To distinguish genders from voices, we need raw data of speech recordings. Our dataset consists of 2 parts.

- We downloaded CMU_ARCTIC speech synthesis databases from Language Technologies Institute at Carnegie Mellon University. We selected 4532 utterances

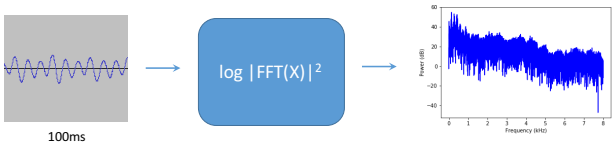

Figure 1: Frequency Spectrogram Generation

speaking out-of-copyright texts from Project Gutenberg[]. Half of them are pronounced by US males, and another half are pronounced by US females. We also downloaded a dataset of 1133 English utterances spoken by a Scottish male, for the purpose of researching into dialect influences. We only selected distributions of 16KHz waveform.

- To create more challenging experiment circumstances, we also collected a small dataset of 30 Chinese utterances with lower signal-to-noise ratio (SNR) from 4 classmates, two of which are males. The average length of these recordings are around 4 seconds. We tested our classifiers on this more challenging dataset, and compared with human performances.

Acoustic Measurements

There are some important acoustic cues of human voices and speeches, with an analyzed frequency range of 0hz-280hz (human vocal range). Intuitively, we care about the mean frequency and standard deviation of frequency. The minimum and maximum fundamental frequencies might also count. Here we proposed a list of 20 acoustic properties as basic classification features, which can be acquired in R using the seewave and tuneR packages. Table 1 illustrates the 20 acoustic measurements acquired from the R package.

Frequency Spectrogram

Since frequency is the critical indication of human voice properties, we also consider encoding frequency information into spectrum to better reveal across-gender differences. A simple 1-D signal from raw audios is illustrated in Figure 1. The typical sampling rates for human speeches are 8KHz and 16KHz, as according to NyquistShannon sampling theorem. Human vocal range is from 0 to 280hz, and thus a sampling rate of 16KHz is guaranteed to capture all the information from a continuous-time audio signal of finite bandwidth. We use simple spectrogram to preprocess the raw data. We split the audio into small windows of unit length (100ms), compute Fast Fourier Transformation (FFT) on the small audio clip and take magnitude. In this way, we can fully capture frequency contents in local window, as illustrated in Figure 1. Finally frames are concatenated from adjacent windows to form "spectrogram" for each raw audio. Each pair of {spectrogram, label} is considered a training data point for our supervised learning model.

However, a critical problem occurs when we consider about the dimension of created features:

$$\begin{aligned} dim(feature) &= (\#audioClips, \#frequencyBins) \\ &= (\frac{len(audio)}{len(audioClip)}, \frac{max(frequency)}{2}) \end{aligned}$$

When the unit length of each audio clip and the maximum of frequency spectrum are fixed, the first dimension of feature matrices varies with respect to the length of input audio data. This would cause a problem for most supervised learning models. To tackle this issue, we use recurrent neural networks for training feature matrices with various dimensions.

Dimension Reduction

From our prior knowledge, when analyzing acoustic features, mean fundamental frequency and standard deviation are already critical cues for distinguish between genders. Thus redundant information might exist in our high-dimensional feature space and impede our learning process. So we utilized a commonly used dimension reduction technique - PCA, to transform our features into low-dimensional space. We first calculate the covariance matrix of the data and compute the eigen vectors on this matrix. The eigen vectors that correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data. They can capture the major characteristics of the whole audio sample. We want to compare the classification performance before and after feature dimension compression, and figure out the trade off between efficiency and accuracy.

Computational Models

Classification is a typical problem in machine learning, which aims at developing an algorithm (classifier) that maps input features to a specific category. A large number of algorithms for classification can be phrased in terms of a linear function that assigns a score to each possible category k by combining the feature vectors of an instance with a vector of weights, such as logistic regression or SVM. We can concert them into non-linear classifiers by applying non-linear kernels. There are also methods that model conditional density functions for discrimination, such as naive Bayes classifier, which belong to the generative model category.

Classic Supervised Learning Models

SVM and Logistic SVM aims at constructing a hyperplane in high-dimensional space to classify data. It can be treated as an constrained optimization problem:

$$\min ||\vec{w}|| \tag{1}$$

$$s.t. \quad y_i(\vec{w}\vec{x_i} - b) >= 1 \quad for \quad i = 1, ..., n \tag{2}$$

We tried with not only linear kernels, but also non-linear kernels of polynomial and sigmoid types. So the problem turns into find the classification vector $\vec{w} = \sum_{i=1}^{n} c_i y_i \Phi(\vec{x_i})$,

Table 1: Error Rates with Frequency Spectrogram[1]

Variable	Definition	Variable	Definition
meanfreq	mean frequency (kHz)	sd	standard deviation of frequency
median	median frequency (in kHz)	Q25	first quantile (in kHz)
Q75	third quantile (in kHz)	IQR	interquantile range (in kHz)
skew	skewness	kurt	kurtosis
sp.ent	spectral entropy	sfm	spectral flatness
Variable	Definition	Variable	Definition
mode	mode frequency	centroid	frequency centroid
meanfun	average of fundamental frequency	minfun	minimum fundamental frequency
maxfun	maximum fundamental frequency	meandom	average of dominant frequency
mindom	minimum of dominant frequency	maxdom	maximum of dominant frequency
dfrange	range of dominant frequency	modindx	modulation index

where the coefficients c_i are calculated by another optimization problem:

$$\max f(c_1,...,c_n) \tag{3}$$

$$= \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i (\Phi(\overrightarrow{x_i})\Phi(\overrightarrow{x_j}))y_i c_i \tag{4}$$

$$= \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i k(\overrightarrow{x_i},\overrightarrow{x_j})y_i c_i \tag{5}$$

$$s.t. \quad \sum_{i=1}^{n} y_i c_i = 0, \quad and \quad 0 \le c_i \le \frac{1}{2n\lambda}, \quad \forall i. \tag{6}$$

Logistic regression is a linear regression model on categorical dependent variables. In our problem, independent variables are feature vectors or matrices extracted from raw audio data, dependent variables are male (level = 1) or female (level = 0). When doing logistic regression, we are actually looking for a curve that best fits the data of the probability of the recording sample belonging to a male vs. its acoustic features. Let the linear model parameters learned by the learning phase to be ($\overrightarrow{w}$,b), where $\overrightarrow{w}$ is a coefficient vector of dimension $1 * p$, the probability of the voice belonging to a male is:

$$P(c_k = male) = \frac{1}{1 + exp(-(\overrightarrow{w}\overrightarrow{x} + b))} \tag{7}$$

We set the threshold to be 0.5 as this is a symmetric classification problem with even prior distribution.

Naive Bayesian We use Bayesian inference to learn parametric models from our training data. Because Bayes classifiers are highly scalable and requiring a number of parameters, we need to train with large number of predictors in the learning process. The basic idea is to learn a parametric likelihood probability model from existing data. Since the overall proportion of male and female is close to 1:1, the prior probability is assumed to be $P(male) = 0.5$ and $P(female) = 0.5$.

[1]From R document for Package WarbleR

We have two basic assumptions for naive Bayes classifier: 1) independence of the predictor variables, and 2) Gaussian distribution (given the target class) of metric predictors.

We formulate our problem as follows: Given a voice instance to be classified, represented by a feature vector $\mathbf{x} = (x_1, x_2, ...x_p)$ which means that there are p independent predictors. So the posterior probability that we want to learn from the training data is:

$$P(c_k | x_1, x_2, ...x_p) \tag{8}$$

where $c_k, k = 1,2$ means the classification outcome for each class. Here $\{c_k\} = \{male, female\}$ in our voice gender categorization problem, $p = 20$ represents the 20 features that we extracted from voice recordings. For continuous data, Gaussian distribution is a typical assumption. For each attribute x_i, we first segment the training data by class c, and let (μ_c, σ_c^2) be the mean and variance of the values in x_i associated with class c. For each observation $\mathbf{x}_j = v$ where $j = 1,2,...N$, the probability distribution of $\mathbf{x}_j$ with value v given a class c, $p(\mathbf{x}_j = v|c)$, can be calculated by a normal distribution parameterized by mean and variance (μ_c, σ_c^2):

$$p(\mathbf{x}_j = v|c) = N(\mathbf{x}_j; \mu_c, \sigma_c^2) \tag{9}$$

And we choose maximum posterior Bayesian decision rule, the final class label $\hat{y} = C_k$ is selected by the following formul

$$\hat{y} = \arg\max_{k=1,2} p(c_k) \prod_{i=1}^{N} p(\mathbf{x}_j | c_k) \tag{10}$$

where N is the number of training samples.

Another critical issue to be solved is how to estimate the parameter pair (μ_c, σ_c^2) for each class c. We also use Bayesian inference. From assumption 1), since each predictor is independent from each other (although this might not be the truth, which we will discuss about later in dimension reduction section), we consider about the 20 predictors separately, and thus the final likelihood numerator of $P(x_1, x_2 ... x_{N_c} | c_k)$ is a product of likelihood probabilities of all the individual predictors

$P(x_i|c_k)$. Also from assumption 2), the functional form for the likelihood of each individual predictor $x_i, i = 1,2,...20$ given a certain class c is a normal distribution:

$$P(x_i|\mu_c,\sigma_c^2) = N(x_i;\mu_c,\sigma_c^2) = \frac{1}{\sigma\sqrt[2]{2\pi}} e^{-((x_i-\mu)^2/2\sigma^2)} \quad (11)$$

So our goal is to infer (μ_c, σ_c^2) based on previous observations. We can estimate μ_c and σ_c^2 separately. To obtain the posterior distribution on μ_c, given all the observations that belongs to class c from our training samples, we can get:

$$P(\mu_c|x_{1i},x_{2i},...x_{N_ci}) \propto P(x_{1i}|\mu_c,\sigma_c^2)P(x_{2i}|\mu_c,\sigma_c^2)...P(x_{N_ci}|\mu_c,\sigma_c^2)P(\mu_c) \quad (12)$$

We assume the prior distribution for μ_c is uniform, i.e., $P(\mu_c) \propto 1$, so the posterior of parameter μ_c is proportional to the product of all the likelihoods for each observation in category c. Then we use expected value the $E(\mu_c)$ as the mean parameter of Gaussian distribution for class c. Similarly, we can estimate σ_c^2 from all the training observations.

Recurrent Neural Network

Speech can be seen as a stream of data, whether in its raw form or feature form. Therefore, we are interested in classes of models that can handle this kind of temporal sequence. Hidden Markov models (HMMs) are traditionally used in speech recognition systems, and have been proved quite successful (Rabiner, 1989). But in our case of gender classification, it is unclear what the hidden states should correspond to. In addition, we suspect that a discriminative model may be sufficient for this binary classification problem, as opposed to a generative model like HMM.

We therefore turn to recurrent neural networks (RNNs). RNNs belong to the broader concept of deep learning (LeCun, Bengio, & Hinton, 2015), where the idea is to learn a mapping from input to output parameterized by multiple layers of simple multiplication and addition operations. These layers learn different levels of abstractions summarizing the input data. Learning is done by backpropagation (Rumelhart, Hinton, & Williams, 1988), which is taking derivatives with respect to individual parameters, while taking advantage of both the chain rule of Calculus and the layer-wise structure.

Recurrent neural networks are discriminative models that map input data sequence of any length $\mathbf{x}_1,\ldots,\mathbf{x}_t \in \mathbb{R}^m$ into a fixed-length vector representation $\mathbf{h}_t \in \mathbb{R}^n$. RNNs consist of two components: the hidden-to-hidden dynamics $W_h \in \mathbb{R}^{n\times n}$ and the input-to-hidden dynamics $W_i \in \mathbb{R}^{n\times m}$, each parameterized with parameters that are shared across all time steps. The simplest form of RNN is given by

$$\mathbf{h}_t = \sigma(W_i\mathbf{x}_t + W_h\mathbf{h}_{t-1} + \mathbf{b}) \quad (13)$$

where σ is some non-linear function (usually the sigmoid) and $\mathbf{b}$ is the bias term.

However, the vanilla RNN model defined above suffers from problems such as gradient vanishing/exploding and is rarely used in practice. Much more practical models now widely used include Long Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho, Van Merriënboer, Bahdanau, & Bengio, 2014). Here we introduce the LSTM formulation, which is used in our experiments. LSTM is a variant of RNN that introduces extra gating operations in the hidden-to-hidden dynamics. There are three gates: input gates $\mathbf{i}$, output gates $\mathbf{o}$, and forget gates $\mathbf{f}$, and two cells: hidden cell $\mathbf{h}$ and memory cell $\mathbf{c}$. The following equations give the dynamics:

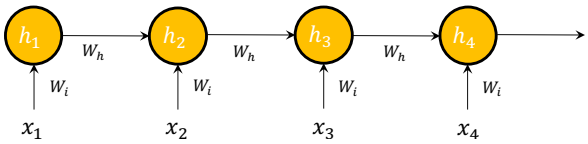

Figure 2: Illustration of Recurrent Neural Network dynamics.

$$\mathbf{i}_t = \sigma(W_i\mathbf{x}_t + U_i\mathbf{h}_{t-1} + \mathbf{b}_i) \quad (14)$$

$$\mathbf{f}_t = \sigma(W_f\mathbf{x}_t + U_f\mathbf{h}_{t-1} + \mathbf{b}_f) \quad (15)$$

$$\mathbf{c}_t = \mathbf{f}_t\mathbf{c}_{t-1} + \mathbf{i}_t\tanh(W_c\mathbf{x}_t + U_c\mathbf{h}_{t-1} + \mathbf{b}_c) \quad (16)$$

$$\mathbf{o}_t = \sigma(W_o\mathbf{x}_t + U_o\mathbf{h}_{t-1} + \mathbf{b}_o) \quad (17)$$

$$\mathbf{h}_t = \mathbf{o}_t\tanh(\mathbf{c}_t) \quad (18)$$

Intuitively, the memory cell $\mathbf{c}$ serves to memorize the relevant information that LSTM has seen so far. At each time step, it preserves some of its previous belief (regulated by the forget gate $\mathbf{f}$, and takes in some of the new observation (regulated by the input gate $\mathbf{i}$. The hidden cell $\mathbf{h}$ is determined by the memory cell, regulated by the output gate $\mathbf{o}$.

Recurrent neural networks have recently been proved successful in many different applications. For example, in image captioning, models proposed in (Vinyals, Toshev, Bengio, & Erhan, 2015; Mao et al., 2014; Donahue et al., 2015) augmented the LSTM with image features, and generates descriptions one word at a time. In visual question answering, models proposed in (Antol et al., 2015) used the LSTM as an encoder and mapped the question into a vector representation, which then interacts with the image features to produce answers. In both cases, the input features $\mathbf{x}$ are word embeddings of individual words.

RNNs have recently proved successful in speech recognition as well (Hannun et al., 2014; Amodei et al., 2016; Povey et al., 2011). What people usually do is to segment the input sound wave into segments of fixed length (e.g. 100ms), run FFT on individual segments, and use the (local) spectrogram as input features $\mathbf{x}$. In our problem, we employed the same approach, and used the hidden state of LSTM after processing the entire sound wave as the feature vector $\mathbf{h}_T \in \mathbb{R}^n$. We simply learn one fully connected layer on top, i.e.

$$P(male) = \frac{1}{1+\exp(-(W_{fc}\mathbf{h}_T + b_{fc}))} \quad (19)$$

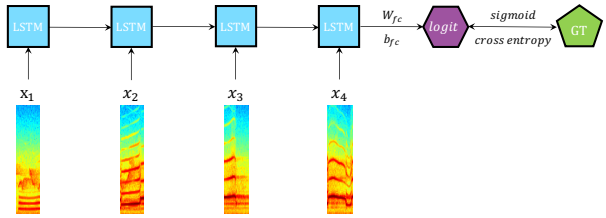

Figure 3: LSTM + fully connected layer model for neural gender classification.

The whole model (see Figure 3) is trained in an end-to-end fashion, using backpropagation through time.

Experiments

To conduct this study, a male and a female corpus are necessary. We shuffled the 4532 utterances of both genders from CMU_ARCTIC speech synthesis databases and split them into training and testing sets by a split factor of 0.6 (2720 training samples and 1812 testing samples).

Acoustic Measurements

We first apply a feature engineering approach to extract 20 critical acoustic measurements from raw data, to form a feature vector of 20 predictors. The concrete meaning for each predictor is already shown in Table 1. Then we experimented with 4 different models: SVM, logistic regression, naive Bayesian, and RNN. The training and testing errors are shown in Table 2. For SVM classification, we used 3 different kernels: linear, sigmoid and polynomial (max degree = 3). The running time for polynomial kernel is much longer than the other two. From Table 2 we can clearly see that linear kernel fits this problem better.

Logistic regression performs very well with this feature selection technique. We chose binomial family for response type, and used cross validation for selecting the best regularizing parameter lambda. Theoretically logistic regression is a simplified deep neural network with single-layer neurons, and thus it achieves excellent performances on such well-defined categorization tasks.

Naive Bayesian model performs well on the low-dimensional feature space. We assume Gaussian distributions for the 20 independent predictors. The parameter pairs for each Gaussian distribution of the predictors are demonstrated in Appendix A.

Then we perform principle component analysis to the 20-dimensional feature space for dimension reduction. The scree plot of importances of principal components (PCs) is visualized in Figure 4. As demonstrated in the figure, it is obvious that there is a significant principal component that accounts for the major projection direction to classify samples. We use the coordinates of data projected on the first 6 principal components, which contribute to 99.99% of total variances.

Table 2: Error Rates with Acoustic Features

Error Type	Training Error(%)	Testing Error(%)
SVM linear	7.76	8.62
SVM sigmoid	53.34	54.23
SVM poly	46.64	46.27
Logistic	2.21	3.26
Bayesian	9.61	9.67
RNN	0	0.31

Table 3: Error Rates with Acoustic Features after PCA

Error Type	Training Error(%)	Testing Error(%)
SVM linear	25.75	25.66
SVM sigmoid	47.00	41.54
SVM poly	47.00	41.54
Logistic	27.28	27.97
Bayesian	30.49	28.28

However, it is clearly seen from the Table 3 that after dimension reduction, the training and testing errors increased significantly compared to previous experiment, except for SVM model with sigmoid kernel.

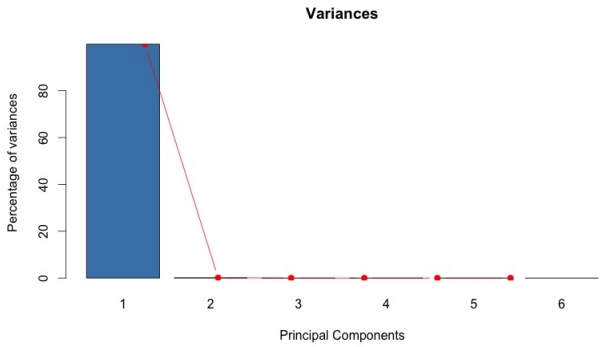

Figure 4: PCs for Acoustic Measurements

Frequency Spectrogram

Then we tried with another feature engineering method: frequency spectrogram generation. We use a unit length of 100ms for each audio clip. Since the sampling frequency in our datasets is 16KHz, we have 1600 sampling points in each unit length recording, and we set 800 frequency bins for the clip. Take a 4-second audio sample as an example, we have 40 audio clips with a 1-by-800 feature vector each, and thus a feature matrix of 40-by-800 is generated for this audio sample. We feed these feature matrices into recurrent neutral networks to learn a classification model. Our implementation is based on Tensorflow (Abadi et al., 2016). The number of nodes in the LSTM is set to 500. We use the Adam optimizer

Table 4: Error Rates with Frequency Spectrogram

Error Type	Training Error(%)	Testing Error(%)
SVM linear	0	0
SVM sigmoid	48.00	49.54
SVM poly	0	0
Logistic	0	0
Bayesian	7.09	7.44
RNN	0	0

(Kingma & Ba, 2014) with a fixed learning rate of 0.00001. The training loss for two 2 epochs is illustrated in Figure 5. The training process converges after almost 3000 iterations. Training and testing errors are also decreasing with the decrement of training loss and converges very closely to 0.

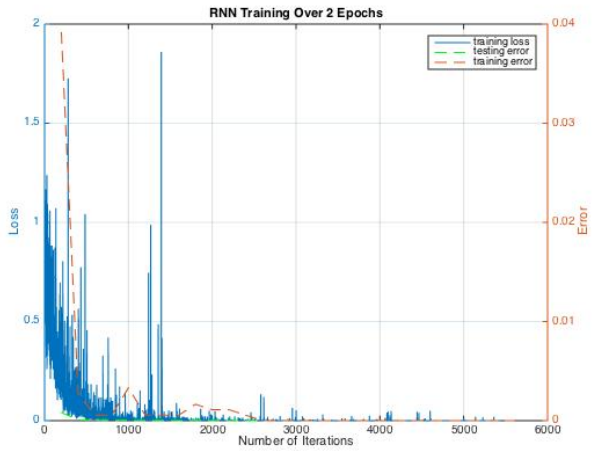

Figure 5: Experiment with Frequency Spectrogram

We also compared the performance of RNN with other categorization models (SVM, logistic, Bayesian) on this feature extraction approach. Since typical classification models do not take features with diverse dimensions, we calculate the 1-by-800 mean frequency spectrum on the N-by-800 feature matrix (on axis = 1), to represent each audio sample. Table 4 illustrates the comparison results. From the table we can see that frequency spectrogram is a powerful feature engineering approach on various computational models except SVM with sigmoid kernel, since it captures all the frequency information of a raw audio sample.

We also do a principal component analysis on the 1-by-800 feature vector, and the first 10 PCs account for 95.19% of total variances. The scree plot of importances of principal components (PCs) of frequency spectrum is visualized in Figure 6. So we selected the first 10 PCs and used the coordinates of data projected on them as the new feature vector. We also compared the performances of classic computational models with features after dimension reduction. From Table 5 we can conclude that after PCA the accuracies did not decrease significantly, which implies that we can achieve a balance between training efficiency and accuracy.

Table 5: Error Rates with Frequency Spectrogram after PCA

Error Type	Training Error(%)	Testing Error(%)
SVM linear	0	0
SVM sigmoid	21.54	20.34
SVM poly	0	0.22
Logistic	0	0
Bayesian	0.33	0.61

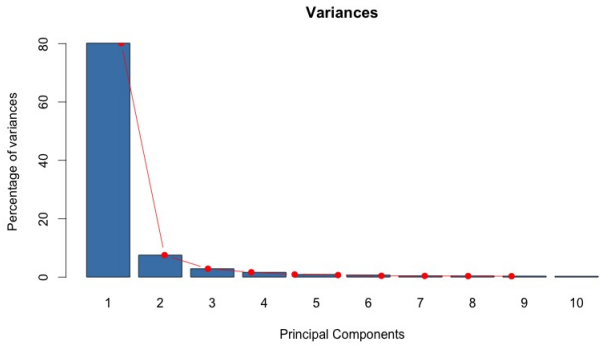

Figure 6: PCs for Acoustic Measurements

More Challenging Circumstances

We also tested our classifier trained by RNN on a smaller dataset recorded in noisy circumstances and compared the accuracy with human recognition performances.

Of the 30 audio recordings, half are spoken by male students, and the other half by female students. Our RNN classifier recognized 9 out of 15 male voice samples, and 8 out of 15 female recordings, while humans recognized 13 out of 15 male and 14 out of female. It is clear that humans are more experienced at recognizing human voice genders than our classifier. Another reason might be that our training data is from CMU_ARCTIC speech synthesis databases, which are recorded in quiet environment by fluent English speakers. We should add more data with lower SNR for training.

Conclusion

In conclusion, this paper aims at recognize gender according to voice and speech. Our focus in on acoustic measurements and frequency spectrum. We experimented with different computational classification models and various feature engineering approaches, applied our RNN classifier to more challenging datasets and compared with human performances. We still need to improve our classification performances by feeding the recurrent neural network with more noisy training data.

Acknowledgments

The author would like to thank Prof. Lu for suggestions and providing relative domain knowledge.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... others (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., ... others (2016). Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning* (pp. 173–182).

Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., & Parikh, D. (2015). Vqa: Visual question answering. In *Proceedings of the ieee international conference on computer vision* (pp. 2425–2433).

Chen, S. H. (2005). The effects of tones on speaking frequency and intensity ranges in mandarin and min dialects. *The Journal of the Acoustical Society of America*, *117*(5), 3225–3230.

Childers, D., Wu, K., Bae, K., & Hicks, D. (1988). Automatic recognition of gender by voice. In *Acoustics, speech, and signal processing, 1988. icassp-88., 1988 international conference on* (pp. 603–606).

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 2625–2634).

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., ... others (2014). Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Johnson, K. (2008). 15 speaker normalization in speech perception. *The handbook of speech perception*, 363.

Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., & Yuille, A. (2014). Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*.

Meena, K., Subramaniam, K. R., & Gomathy, M. (2013). Gender classification in speech recognition using fuzzy logic and neural network. *Int. Arab J. Inf. Technol.*, *10*(5), 477–485.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., ... others (2011). The kaldi speech recognition toolkit. In *Ieee 2011 workshop on automatic speech recognition and understanding*.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Rose, P. (1991). How effective are long term mean and standard deviation as normalisation parameters for tonal fundamental frequency? *Speech Communication*, *10*(3), 229–247.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, *5*(3), 1.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 3156–3164).

Zeng, Y.-M., Wu, Z.-Y., Falk, T., & Chan, W.-Y. (2006). Robust gmm based gender classification using pitch and rasta-plp parameters of speech. In *Machine learning and cybernetics, 2006 international conference on* (pp. 3376–3379).