Reconhecimento de Locutor

Ramon Duarte de Melo & André Ribeiro Queiroz

Universidade Federal do Rio de Janeiro

ramonduarte@poli.ufrj.br & handre_queiroz@poli.ufrj.br

8 de maio de 2019

Sumário

Objetivo

Usar biometria de voz para identificar quem está falando

O Sinal de Voz

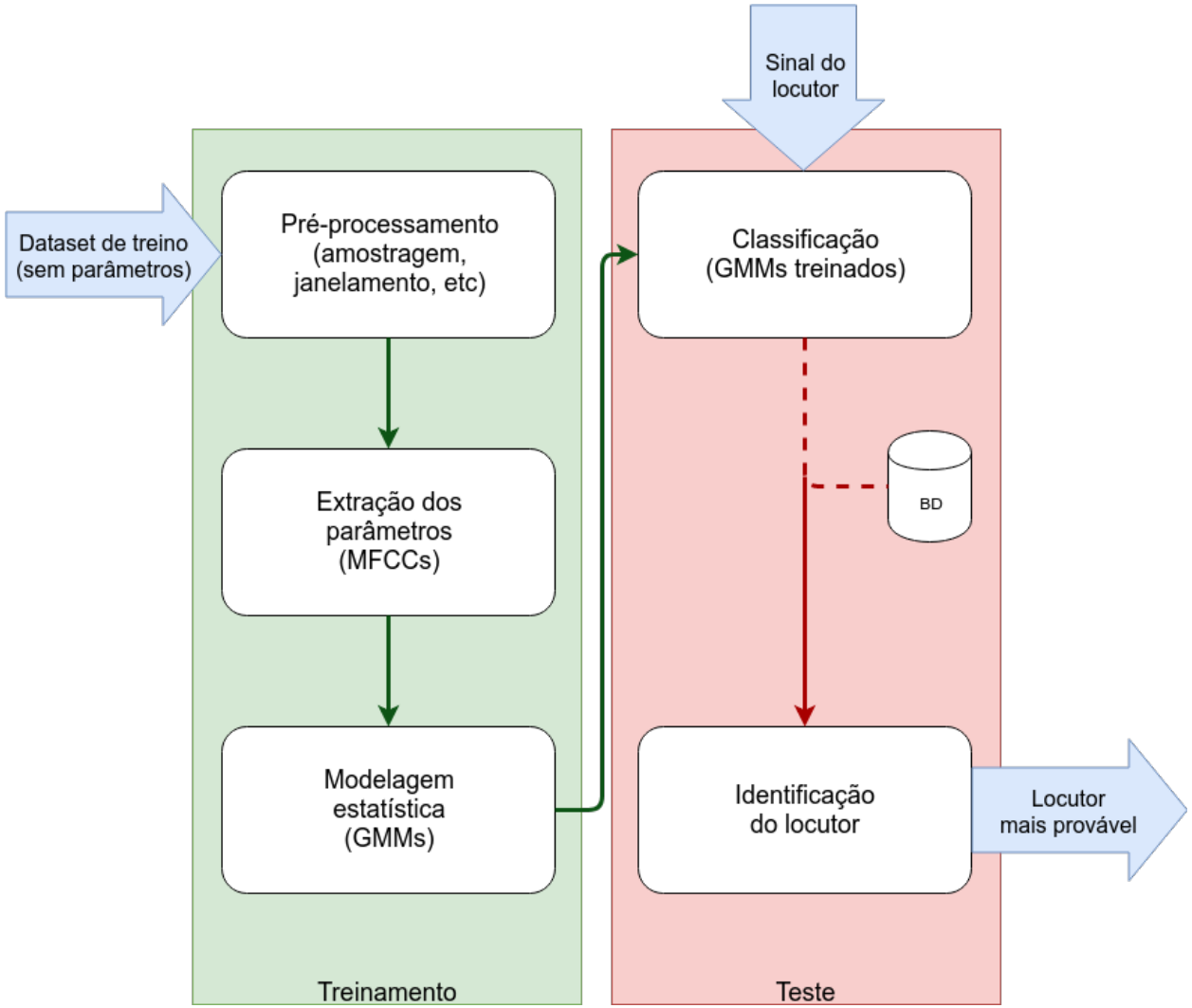
O sinal de áudio recebido como entrada é produzido através de diversas transformações que ocorrem em estágios distintos da captação sonora. Portanto, precisamos extrair somente as propriedades sonoras chamadas de "dependentes do locutor".

Este processamento é geralmente feito por parametrização, como o LPC (*Linear Prediction Coding*):

$$s(n) = -\sum_{k=1}^{p} a_k \cdot s(n-k) + e(n)$$

- p é a ordem de predição;
- a_k são os coeficientes de predição
- $s(n-k)$ são ps sinais anteriores
- $e(n)$ é o erro de predição.

Visão Geral do Sistema



Captação do sinal

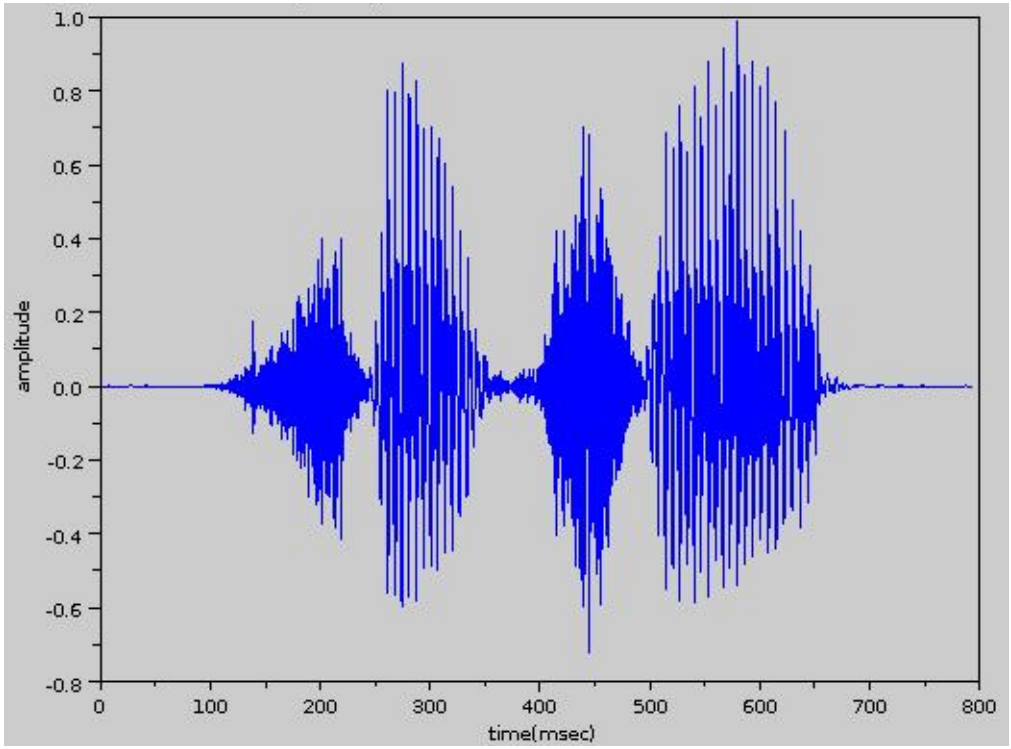
- Áudios de entrada amostrados em 8kHz ou 16kHz
- 16 bits por amostra
- Duração variável entre 10 e 60 segundos

Datasets

Dataset de treinamento: 34 locutores, 5 frases para cada locutor, 20-30 segundos
Dataset de teste: os mesmos 34 locutores, 1 frase para cada, $\approx$10 segundos

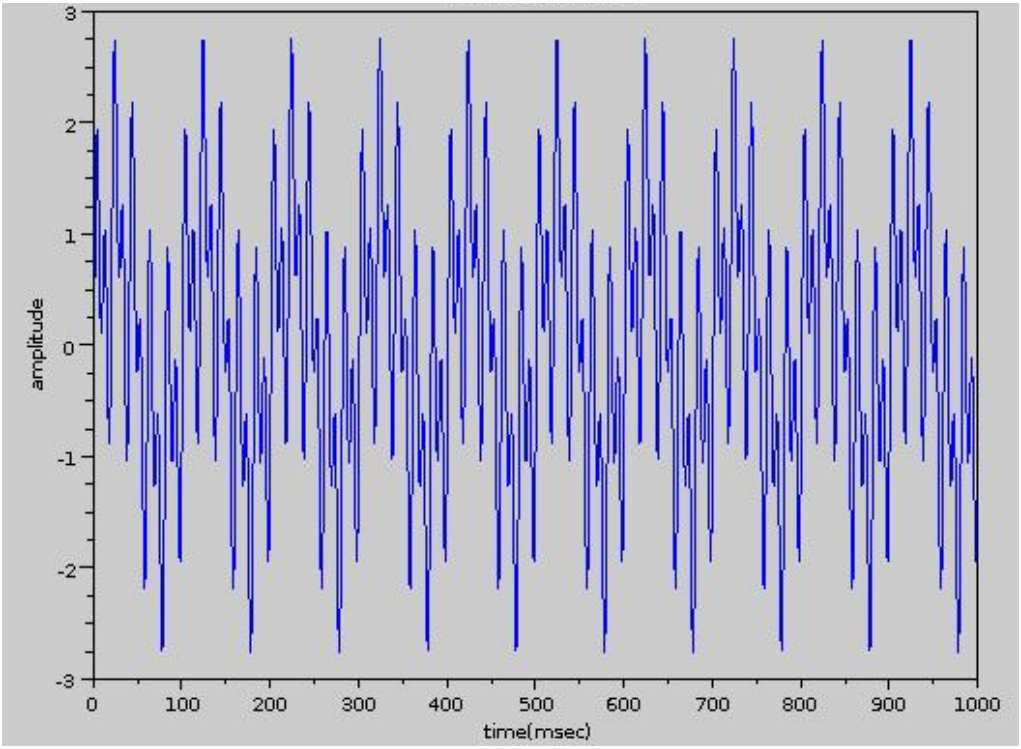
Amostragem

O sinal de voz é tradicionalmente não-estacionário, o que dificulta a análise das frequências contidas nele.



Amostragem

O sinal de áudio é dividido em quadros de 20-30 ms para encontrar estacionariedade.

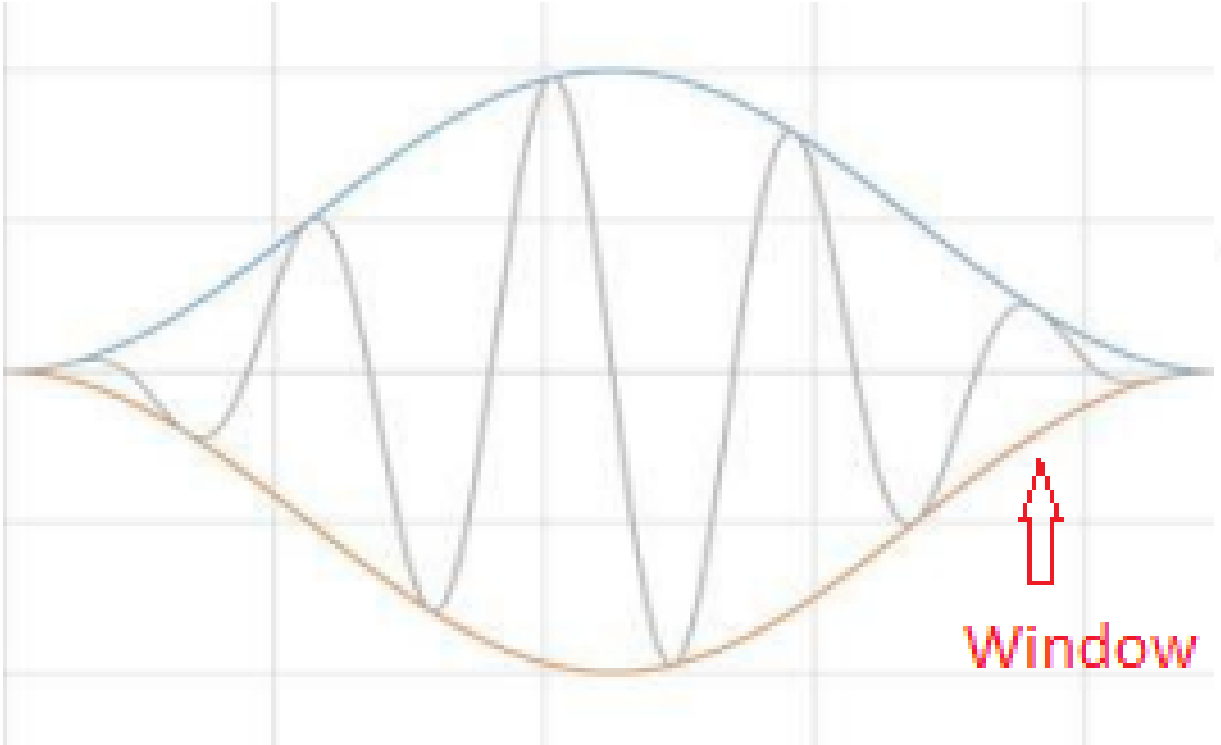


Janelamento

Como o intervalo de amostragem é arbitrário, o sinal apresenta descontinuidades nas extremidades da onda extraída.

Janelamento

Para isso, é feito um janelamento que reduz as extremidades gradativamente a zero.



Parametrização

A extração dos parâmetros é feita pela técnica MFC (*Mel-frequency cepstrum*):

1. Janelamento do sinal
2. Transformada de Fourier das janelas
3. Mapeamento das potências do espectro na escala de Mel
4. Logaritmos das potências nas frequências de Mel
5. Transformada discreta de cosseno dos logaritmos
6. Obtenção das amplitudes do espectro resultante (são os MFCCs)

Modelagem

Pelo MFC, são extraídos 40 coeficientes (chamados de MFCCs), que alimentam GMMs (*modelos misturados de Gauss*).

Durante o treinamento, os GMMs tentam aprender a distribuição dos MFCCs do dataset de treino.

Durante o teste, os GMMs darão "notas"aos MFCCs do dataset de teste.

Modelagem

GMMs são modelos que agrupam linearmente k distribuições normais para tentarem imitar a distribuição probabilística das subpopulações. Por isso, a identificação dos locutores consiste no locutor mais provável.

Essa probabilidade $P(X|\lambda)$ é dada por:

$$P(X|\lambda) = \sum_{k=1}^{K} \omega_k P_k(X|\mu_k, \sum_k)$$

onde $P_k(X|\mu_k, \sum_k)$ é a distribuição normal:

$$P_k(X|\mu_k, \sum_k) = (\sqrt{2\pi|\sum_k|})^{-1} e^{1/2(X-\mu_k)^T \sum^{-1}(X-\mu_k)}$$

Módulos

Existem várias ferramentas que calculam MFCCs. Optamos pelo python_speech_features:

```
import python_speech_features as mfc
from sklearn import preprocessing
def mfcc(sinal, taxa=16000, janela=0.03, dist_janelas=
                                0.01):
    parametros = mfc.mfc(sinal, taxa, janela,
                                    dist_janelas)
    parametros = preprocessing.scale(parametros)
    return parametros
```

Módulos

Dos parametros extraídos anteriormente, apreendemos GMMs usando o módulo sklearn:

```
from sklearn.mixture import GMM
gmm = GMM(n_components = 8, n_iter = 200,
                            covariance_type='diag',
                            n_init = 3)
gmm.fit(parametros)
```

Módulos

Em tese, o número de componentes e o total de iterações não deveriam ser arbitrários, mas sim descoberto através do algoritmo k-médias.
Para este trabalho, no entanto, optamos por utilizar os valores padrão:

- $k = 8$: os dados sempre terão 8 agrupamentos (e 8 centroides).
- 200 iterações: maioria dos exemplos utilizavam este valor.

Objetos a serem testados

- Datasets
- Parametrizações
- Modelagens estatísticas
- Ferramentas
- Pipelines