

Reconhecimento de Gêneros de Pessoas a partir dos sinais de suas vozes

Kallil M. Caparroz¹, Rodrigo C. Anater²

¹Universidade Tecnológica Federal do Paraná - Câmpus Pato Branco

kallil@alunos.utfpr.edu.br, rodrigoanater@alunos.utfpr.edu.br

Abstract. *During the past two decades, the Machine Learning technique has become one of the main studies of information technology. With the increasing amount of information to be analyzed, there is a good reason to develop more and more computational algorithms to analyze these data. The objective of this work is to use different algorithms in the same dataset and compare which one is the best to be used.*

Resumo. *Durante as duas últimas décadas, a técnica de Aprendizado de Máquinas tornou-se um dos principais estudos da tecnologia de informação. Com o aumento da quantidade de informações a serem analisadas, há uma boa razão para desenvolver mais e mais algoritmos computacionais a fim de analisar esses dados. O objetivo deste trabalho é usar diferentes algoritmos no mesmo conjunto de dados e comparar qual deles é o melhor a ser usado.*

1. Introdução

O termo *Machine Learning* se refere à detecção, por parte do computador, em reconhecer padrões em uma base de dados. Nas últimas décadas, tornou-se uma ferramenta fundamental para qualquer tarefa que requer a extração de informações de grandes conjuntos de dados.

Estamos cercados por tecnologia baseada em *Machine Learning*: Mecanismos de pesquisa aprendem como nos disponibilizar os melhores resultados, softwares anti-spam aprendem como filtrar nossos e-mails, assim como transações de cartões de créditos são mantidas em segurança por um software que aprende a detectar fraudes. [Shalev-Shwartz and Ben-David 2014] Para tal, vários algoritmos conseguem fazer com que um computador possa automaticamente detectar os padrões de uma base de dados.

Nesse trabalho será utilizada uma base de dados criada para identificar uma voz como masculina ou feminina baseada nas propriedades acústicas da voz e da fala. A base de dados consiste em 3 168 amostras de voz gravadas, coletadas de falantes masculinos e femininos, tendo 22 características, todas envolvendo a frequência sonora da fala, sendo algumas delas:

- Frequência média (kHz)
- Desvio padrão da frequência
- Mediana da frequência (kHz)
- Centróide da frequência

Para o reconhecimento dos padrões dessa base de dados serão utilizados duas técnicas: Regressão Linear e Máquinas de Vetores de Suporte.

2. Regressão Linear

O principal objetivo por trás da regressão linear é de estimar um valor $y \in \mathbb{R}$ dado uma característica x . Por exemplo, pode-se estimar o valor de um estoque no dia seguinte de acordo com as vendas dos dias anteriores, os batimentos cardíacos de um atleta de acordo com a distância que percorreu, entre outras aplicações [Smola and Vishwanathan 2008].

A Figura 1 mostra um bom exemplo de um método de regressão linear. Os pontos na figura são amostras obtidas das características x da base de dados, o objetivo é criar uma função $f(x)$ que se aproxime da melhor forma dos valores observados.

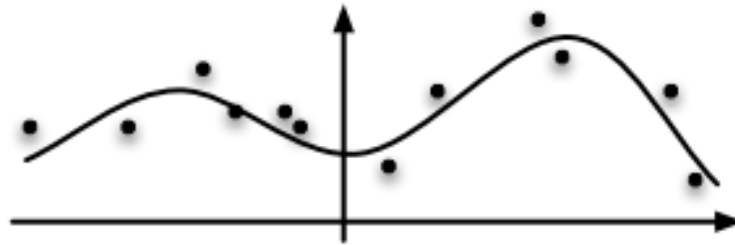


Figura 1. Exemplo de regressão linear

3. Materiais e Métodos

3.1. Materiais

Para captar e realizar a leitura do ruído eletromagnético foram utilizados uma antena para a captação e um microcontrolador para a aquisição do valor do ruído captado.

3.1.1. Arduino UNO

O Arduino Uno (Fig 1) é uma placa de microcontrolador baseado no ATmega328 (datasheet). Possui 14 pinos de entrada/saída digital (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal oscilador de 16MHz, uma conexão USB e uma entrada de alimentação uma conexão ICSP.

O microcontrolador foi responsável pela leitura do ruído eletromagnético em uma de suas portas analógicas. O ruído foi então convertido para um valor numérico através do conversor analógico-digital (ADC) presente no Arduino.

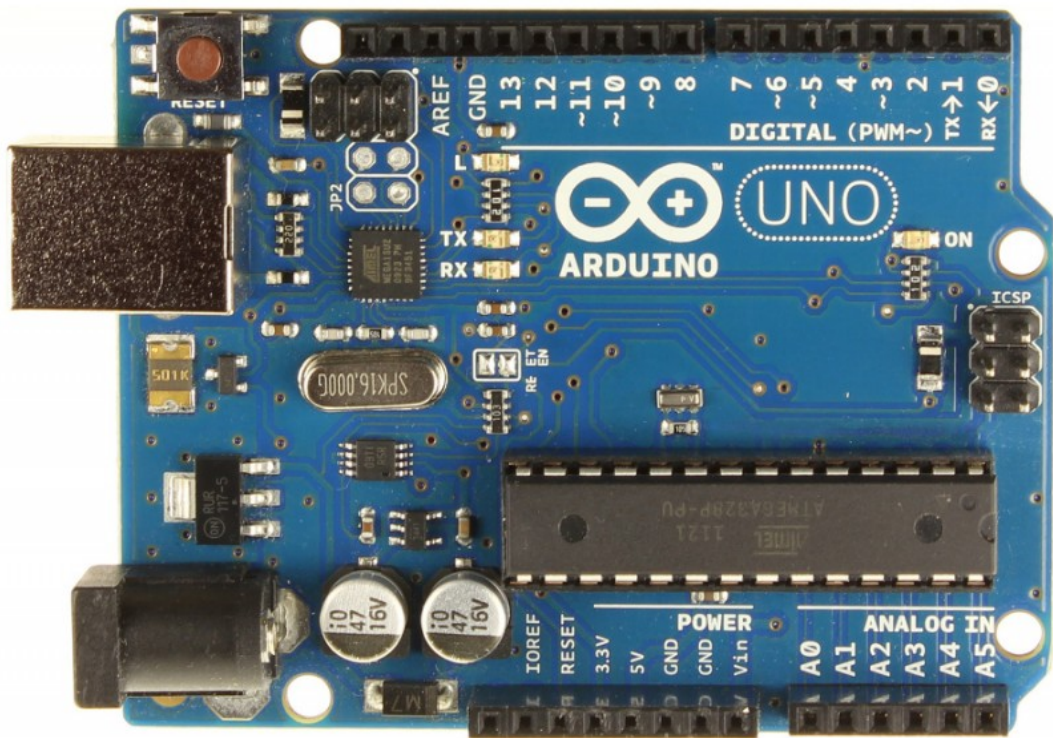


Figura 2. Arduino UNO

3.1.2. Antena

Foi utilizado um fio de estanho na porta do microcontrolador como aparato para a captação de interferência eletromagnética. A configuração escolhida da antena foi de monopolo de quarto de onda [?]. A frequência captada pela antena é prevista pela equação abaixo:

$$f = \frac{c}{\frac{L}{4}} \quad (1)$$

sendo f a frequência em Hertz, c a velocidade da luz e L o comprimento da antena. Para uma antena de 8 centímetros, a frequência captada pela antena é de 15MHz.

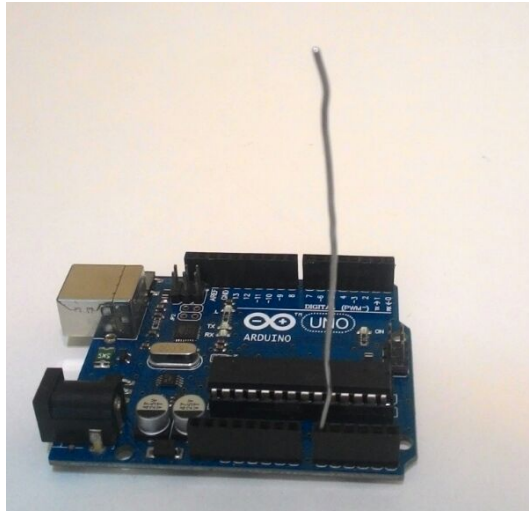


Figura 3. Arduino UNO com a antena monopolo

3.2. Métodos

Os dados adquiridos pela antena foram tratados no próprio microcontrolador. O ruído ocasionou uma pequena variação de tensão na porta analógica do Arduino UNO, que foram convertidos para valores na faixa de 0 até 1023. Foram realizadas dez mil leituras, com um pequeno delay (cerca de 20 milisegundos) entre leituras sucessivas, para evitar ler o mesmo ruído múltiplas vezes. O processamento destas leituras foi feito de acordo com o algoritmo abaixo.

Algoritmo 1: Algoritmo de tratamento de dados

Input: Ruído da porta analógica do microcontrolador

Output: Sequência de números em base hexadecimal

Início

while *existirem números a serem adquiridos* **do**

1. $prev_value \leftarrow$ valor lido da porta do microcontrolador

2. $value \leftarrow$ próximo valor lido da porta do microcontrolador

3. $d \leftarrow |value - prev_value|$

4. $d \leftarrow d \bmod 16$

5. Imprima d na base hexadecimal

end

Este código fornece uma sequência de números na base hexadecimal (0-F). Para as ferramentas de validação do experimento, era necessário um arquivo binário. Para isso, foi utilizado um código em C para ler os números em hexadecimal e transformá-los em uma sequência binária.

```
#include <stdio.h>
```

```
int main(void) {  
    unsigned int a;  
    while(scanf("%X", &a) != -1) {
```

```
    printf("%c%c", a>>8, a);  
}  
return 0;  
}
```

Código 1. Conversão hexa-binário

4. Validação dos resultados

Para avaliar o nível de aleatoriedade das sequências, foi utilizado o programa *ent*, que foi criado com o objetivo de realizar a avaliação de geradores de sequências numéricas pseudoaleatórias para criptografia e aplicações de amostragem estatística. O programa realiza cinco testes diferentes:

- Entropia: Analisa a densidade de informação apresentada, ou seja, verifica o quanto a sequência pode ser comprimida e ainda representar o dado original. Quanto menor for a entropia, menos aleatória e mais compressível é a sequência.
- Teste Qui-quadrado: É calculada a distribuição qui-quadrado, que avalia a relação entre os valores fornecidos e os esperados. A distribuição é simétrica, sendo que sequências verdadeiramente aleatórias devem resultar em valores mais centralizados [?].
- Média Aritmética: Realiza a média aritmética dos valores. Considerando que a análise é feita a partir de blocos de quatro bytes, ou seja, são valores entre zero e 255, idealmente a média deve se aproximar de 127,5.
- Aproximação de π pelo método de Monte Carlo: Cada sequência sucessiva de seis bytes é utilizada como coordenadas em um quadrado, o qual contém um círculo inscrito. Pontos verdadeiramente aleatórios tem uma probabilidade de estar dentro do círculo igual à razão entre a área do círculo pela área do quadrado. Assim, a razão entre os pontos do círculo e o total de pontos deve tender (lentamente) à um quarto do valor de π .
- Coeficiente de correlação serial: Mede o quanto cada elemento da sequência se relaciona ao elemento anterior, e para sequências verdadeiramente aleatórias, deve se aproximar de zero.

Deve-se ressaltar, porém, que não existem testes definitivos de aleatoriedade, uma vez que qualquer teste deve-se basear em probabilidades, e não certezas. Dessa forma, é de se esperar que, após realizada uma certa quantidade de testes, uma sequência verdadeiramente aleatória falhe em alguns deles, porém com menor probabilidade de que uma sequência não aleatória

Referências

- Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.
- Smola, A. and Vishwanathan, S. (2008). Introduction to machine learning. *Cambridge University, UK*, 32:34.