Reconhecimento de Padrões - Relatório Probabilidade e Probabilidade Condicional

Daniel Rosendo de Souza[1]
Instituto Federal de Tecnologia e Ciência (IFCE Campus Maracanaú)

Resumo: Este relatório irá descrever um problema de probabilidade estatística e probabilidade condicional, usando a regra de bayes para solucionar alguns problemas. Mostrar algumas definições vista em sala de aula e aplicando-as com técnicas de programação e por fim mostrando graficamente os resultados.

1. Introdução

Probabilidade é o estudo das chances de obtenção de cada resultado de um experimento aleatório. A essas chances são atribuídos os números reais do intervalo entre 0 e 1. Resultados mais próximos de 1 têm mais chances de ocorrer. Além disso, a probabilidade também pode ser apresentada na forma percentual.A probabilidade condicional refere-se à probabilidade de um evento ocorrer com base em um evento anterior.

Em teoria das probabilidades e estatística, o teorema de Bayes descreve a probabilidade de um evento, baseado em um conhecimento a priori que pode estar relacionado ao evento. O teorema mostra como alterar as probabilidades a priori tendo em vista novas evidências para obter probabilidades a posteriori. Por exemplo o teorema de Bayes pode ser aplicado ao jogo das três portas.

Foi proposto um problema, onde envolve as definições acima, o problema consiste em fazer um programa que consiga carregar um texto, pode ser ele qualquer ou não, onde deve-se fazer três pontos, calcular e plotar a distribuição de probabilidade p(x) sobre as 27 letras x, que é calcular quantas vezes cada letra aparece no texto, calcular e plotar a distribuição de probabilidade conjunta p(x,y) sobre os 27x27 possíveis bi-gramas xy, que é calcular a probabilidade de todas as combinações de letras e por último, calcular e plotar a distribuição condicional.

Todo o trabalho foi feito na linguagem python, usando algumas bibliotecas para as plotagem dos gráficos, pegar as expressões regulares e afins. Esse trabalho será dividido da seguinte forma: Desenvolvimento onde iremos explicar como foi feito o programa e qual ferramenta usamos para desenvolver, Codificação e Resultados, com fotos dos resultados e explicação de trechos de código e por fim uma breve conclusão do que foi aprendido.

2. Desenvolvimento

O projeto foi desenvolvido em python, devido suas facilidades na linguagem, na programação em si, a ferramenta para auxiliar o seu desenvolvimento foi a IDE PyCharm, como toda IDE nos trás alguns benefícios em relação a codificação.

O corpo do projeto foi dividido em 7 funções e 1 função main, onde cada função tem um propósito fundamental para o problema, foi utilizado algumas bibliotecas auxiliares, são elas: A biblioteca string onde é ela que irá fornecer nosso alfabeto com todas as letras, a biblioteca re que nos dá todas as expressões regulares que iremos utilizar para poder tratar

nosso texto, a biblioteca matplotlib a biblioteca mais fundamental de nosso problema pois é ela quem está encarregada de plotar os gráficos com os resultados e por fim utilizamos a biblioteca numpy para podermos fazer algumas transformações em nossas listas.

3. Codificação e Resultados

O nosso primeiro desafio é calcular e plotar a distribuição de probabilidade p(x) sobre as 27 letras. A nossa função primeira questão receberá como parâmetro o texto e nossa lista com o nosso dicionário. Inicialmente devemos abrir o nosso arquivo que contém o texto, colocar ele em um vetor e assim tratar ele, removendo todos os caracteres especiais, como acento, ponto, vírgula e números, pois esses caracteres não fazem parte do alfabeto.

```
def main():
    alphabet = str.ascii_lowercase + ' '
    letters = {}
    bigram = {}

    for letter in alphabet:
        letters[letter] = 0

    arquivo = open('test.txt', 'r')
    vetor_texto = arquivo.read().lower()

    vetor_texto = re.sub('[^a-zA-Z ]', '', re.sub(r'\.', ' ', vetor_texto))
```

Figura 1 - Método Main

Para calcular a probabilidade de uma letra ‘x’ aparecer em um texto, basta pegar a quantidade de vezes que a letra ‘x’ apareceu e dividir pelo total de letras do texto.

```
def primeira_questao(texto, letters):
    totalLetters = total_de_letras(texto, letters)

    for i in letters:
        letters[i] = round(letters[i] / totalLetters, 5)

    return letters
```

Figura 2 - Função Pimeira Questão

Onde nossa variável totalLetters chama a função total de letras que irá me retornar a soma de todas as letras e por fim como resultado temos o seguinte gráfico mostrando a quantidade de ocorrências de cada caractere no texto:

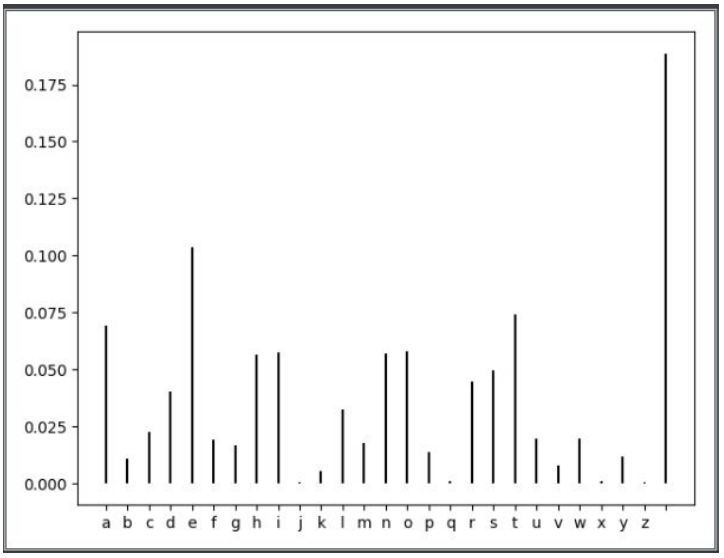

Figura 3 - Gráfico Amostral

Para o segundo problema devemos calcular e plotar a distribuição de probabilidade conjunta p(x,y), sobre os 27x27 possíveis bi-gramas xy. A função segunda questão, irá receber nosso alfabeto, nossa lista de bigramas e nosso texto, já também tratado feito as remoções de caracteres especiais. Um bigrama, podemos imaginar que seja uma matriz, 27 por 27 onde cada linha e coluna é uma letra do alfabeto, e nela iremos analisar todas as combinações de letras como por exemplo aa, ab, ac, ad, bb, ba e assim por diante.

Nossa função segunda questão, com o nosso conjunto de alfabetos irá povoar o nosso bi-grama, com todas as posições da matriz com valor zero, e irá contar a quantidade, analisar, no texto quantos bi-gramas tem. E por fim calcula a probabilidade do bi-grama.

```
def segunda_questao(alphabet, bigram, texto):
    for i in alphabet:
        for j in alphabet:
            bigram[i + j] = 0

    for i in range(len(texto) - 1):
        bigram[texto[i] + texto[i + 1]] += 1
    ##print("O total de bigramas é: ", bigram)

    for i in bigram:
        bigram[i] = round(bigram[i] / (len(texto) - 1), 5)

    ##print("A probabilidade de bigramas é: ", bigram)

    return bigram
```

Figura 4 - Função Segunda Questão

E como resultado temos a plotagem do seguinte gráfico de cores:

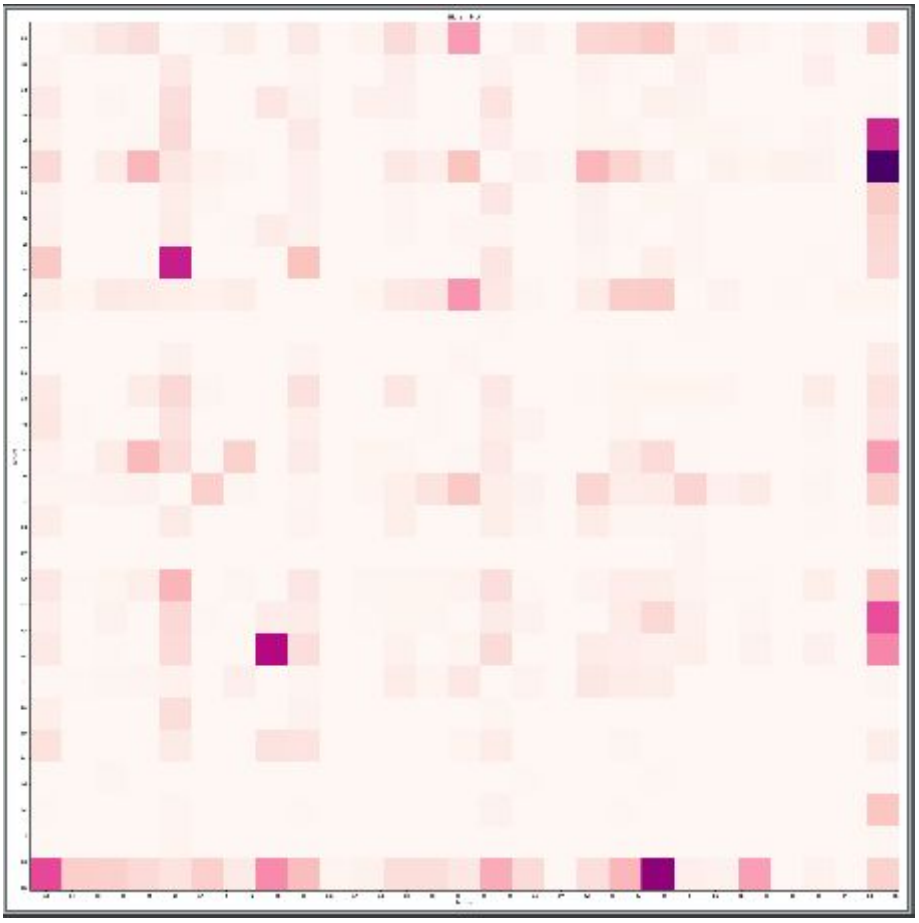

Figura 5 - Gráfico Amostral Bi-grama

E por fim devemos a partir da probabilidade conjunta, calcular e plotar a distribuição condicional:

1 - p(y|x), a distribuição condicional da segunda letra y dada a primeira letra sendo x.

2 - p(x|y), a distribuição condicional da segunda letra x dada a primeira letra sendo y.

Nossa função irá receber como parâmetro o nosso bi-grama, já calculado do item anterior e nossa lista com dicionário, onde essa lista já nos dá o total de cada letra no texto. Temos 2 lista uma para cada condicional pedida, e cada laço de repetição irão calcular nossa distribuição, com relação do bi-grama dividido pela total de letras dada.

```
def terceira_questao(bigram, letters):
    probCond = {}
    probCond2 = {}
    for i in letters:
        for j in letters:
            probCond[i + j] = round(bigram[j + i] / letters[i], 6)

    print("prob x|y: ", probCond)

    for i in letters:
        for j in letters:
            probCond2[i + j] = round(bigram[i + j] / letters[i], 6)

    print("prob y|x ", probCond2)

    plot_graf_sec(probCond)
    plot_graf_sec(probCond2)
```

Figura 6 - Função Terceira Questão

E por fim, temos como resultados os dois gráficos a seguir:

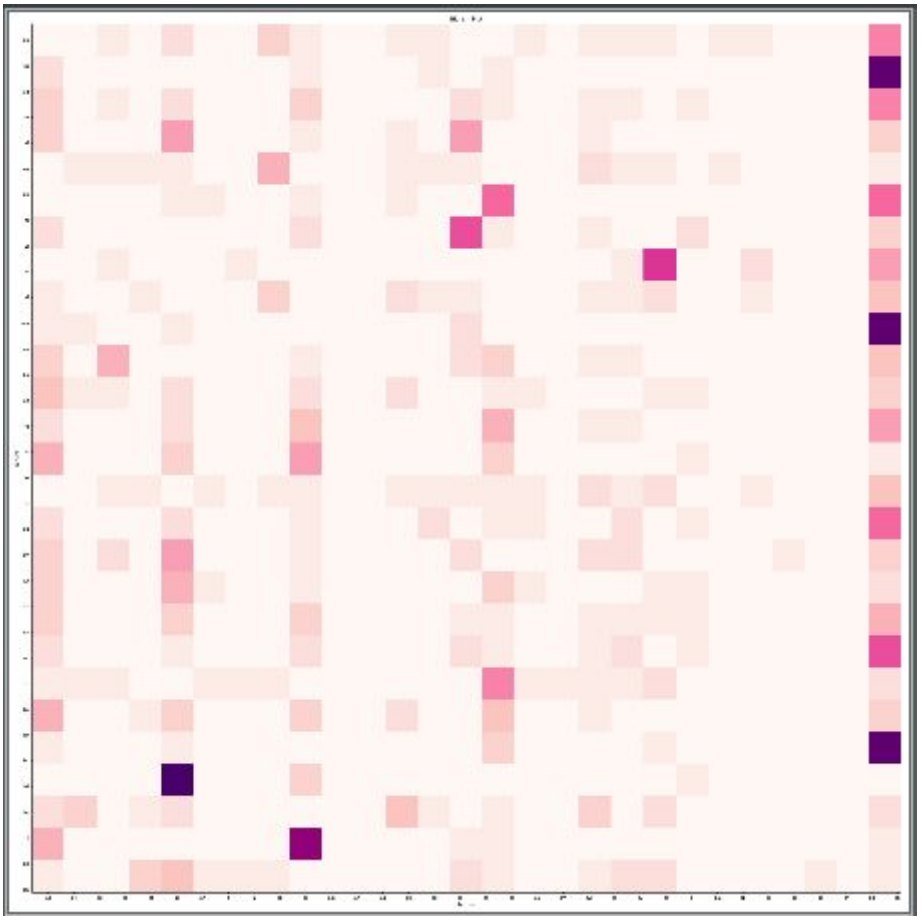

Figura 7 - Gráfico para p(y|x)

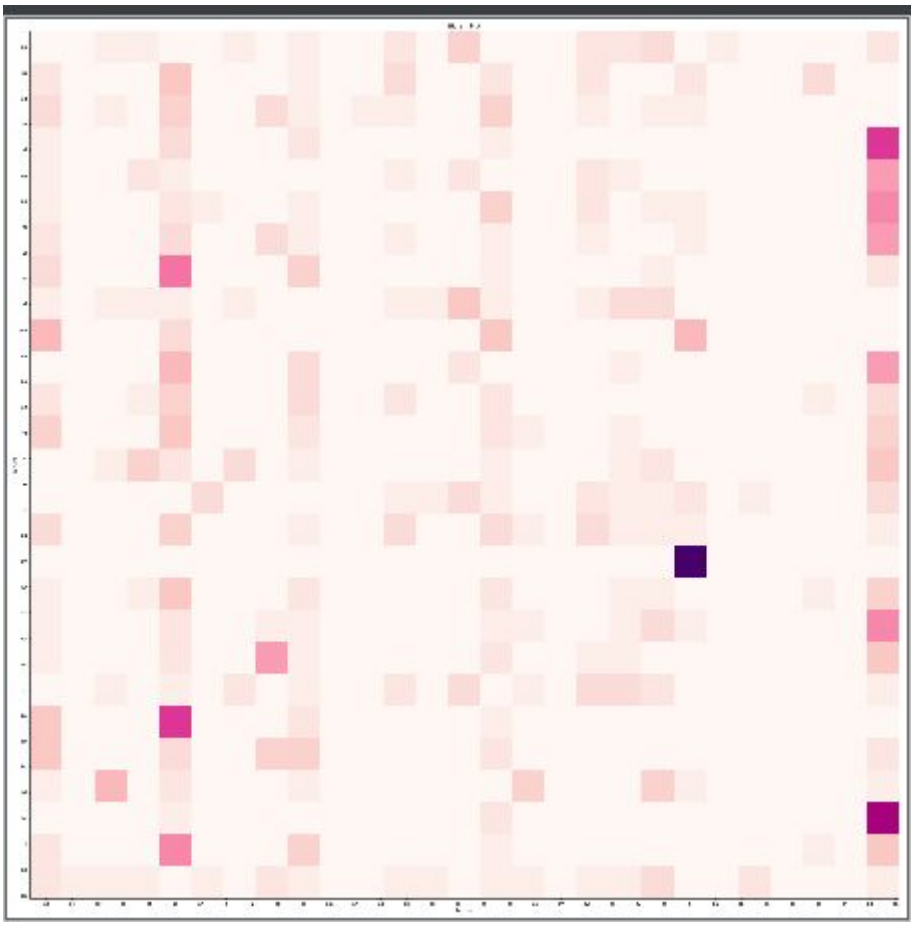

Figura 8 - Gráfico para p(x|y)

4. Conclusão

Podemos concluir neste trabalho que com técnicas computacionais, podemos visualizar melhor as nossas distribuições de probabilidade, tendo uma maior explicação e detalhamento do seu funcionamento.