

Instituto Federal de Educação, Ciência e Tecnologia do Ceará
Engenharia da Computação

Visão Computacional

Prof. Pedro Pedrosa

Relatório N^o 2 - Machine Learning

Mariana Akeme Ogawa

Fortaleza

Abril de 2019

Sumário

1	Introdução	2
2	segmentação	2
2.1	momento de Hu	3
2.2	LBP	3
3	classificadores	3
3.1	Perceptron Simples	3
3.2	kNN	4
4	Resultados	4
5	Conclusão	6

1 Introdução

Este relatório tem como objetivo principal apresentar a extração e classificação de características de uma base. Sendo que essa base é composta por um arquivo .txt no qual cada linha do arquivo é um número de 0 a 9. Cada imagem formada por uma linha do TXT possui dimensões 35x35, sendo cada linha representada em sequencia no TXT, e portanto cada linha do TXT possui 1225 numeros (0 ou 1), e no fim o rótulo a qual representa aquela imagem. Ao formar a imagem, é considerado o valor 0 como preto e 1 como branco. Na figura 1 apresenta a imagem convertida do texto.

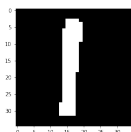


Figura 1: Exemplo de imagem dos números utilizados.

Após aplicar o extrator deve-se utilizar classificadores, sendo que 50% da base deve ser utilizada para o treinamento do classificador e os 50% restantes como teste.

Para a realização desse trabalho foram utilizado as bibliotecas NumPy, Matplotlib e Sklearn para o Python, além da biblioteca de visão computacional OpenCV [1]. Sendo que a biblioteca Sklearn é especializada em computação científica e Machine Learning [2]. Desta forma, foram não foram implementados os algoritmos bases de segmentação e classificação sendo utilizado apenas as funções prontas das bibliotecas supracitadas.

2 segmentação

A segmentação de imagens não triviais é uma das tarefas mais difíceis no processamento de imagens. Por causa dessa dificuldade, existem situações, como nas aplicações de controle industrial, é possível pelo menos alguma medida de controle sobre o ambiente. Porém, em outras aplicações, como a busca autônoma de alvos, não é possível controlar o ambiente no qual o sistema irá operar, nessa situação para auxiliar o processamento digital da imagem deve-se selecionar sensores que provavelmente vão realçar os objetos de interesse, diminuindo a contribuição dos detalhes irrelevantes da imagem. [3]

A maioria dos algoritmos de segmentação baseia-se em uma das seguintes propriedades básicas de valores de intensidade: descontinuidade e similaridade.* Na primeira categoria, a abordagem é dividir uma imagem com base nas mudanças bruscas de intensidade, como as bordas. As abordagens principais na segunda categoria estão baseadas na divisão de

uma imagem em regiões que sejam semelhantes de acordo com um conjunto de critérios predefinidos. A limiarização, o crescimento de região e a divisão e fusão de regiões são exemplos dos métodos desta categoria [3]

2.1 momento de Hu

O momento de Wu é um momento estatístico utilizado como descritor da imagem. Sendo o momento central, mais comum, também chamado de variância e apresenta de que forma a função densidade se “espalha”. [4]

Momentos de ordem mais altas podem conter outras informações do formato da função de densidade, como a torção – tendência de os pontos estarem de um lado da média ao invés do outro. Um importante teorema da probabilidade é que cada momento determina uma densidade única da função.[3]

2.2 LBP

O padrão local binário, LBP (do inglês local binary pattern), sendo derivado da definição geral de textura em uma vizinhança local, sendo definido como uma medida invariante de escala de cinza e é uma ferramenta poderosa para modelar a textura de imagens. O operador original LBP categoriza os pixels de uma imagem utilizando um limiar em uma vizinhança 3×3 comparando com o pixel central e concatenando os resultados binários para formar um número. [3]

3 classificadores

3.1 Perceptron Simples

O perceptron é a forma mais simples de redes neurais e é utilizado para a classificação de padrões que são linearmente separáveis, por exemplo, padrões que se encontram em lados opostos do hiperplano. Basicamente, consiste em um único neurônio com pesos sinápticos e bias(a inclinação) ajustáveis. [5]

O algoritmo utilizado para ajustar os parâmetros livres dessa rede neural foi apresentado em um procedimento de aprendizagem desenvolvido por Rosenblatt, para o modelo do perceptron cerebral. Rosenblatt provou que se os padrões (vetores) utilizados para treinar o perceptron são apresentados como duas classes linearmente separáveis, então o algoritmo converge e posiciona a superfície de decisão na forma de um hiperplano entre as duas classes. A prova de convergência do algoritmo é conhecida como Teorema da Convergência

do Perceptron. Esse neurônio foi construído tendo como base um neurônio não linear, chamado de modelo McCulloch-Pitts. [6]

O perceptron construído a partir de um único neurônio é limitado a fazer a classificação em apenas duas classes. Expandido a camada de saída, computacional, do perceptron para incluir mais de um neurônio, pode-se conseguir classificar em mais de duas classes. Entretanto, as classes devem ser linearmente separáveis para que o perceptron seja efetivo. [6]

3.2 kNN

k-média é um algoritmo de agrupamento simples e de também um algoritmo de segmentação básica. O algoritmo k-médias foi proposto por MacQueen, onde o k representa o número de grupos e são agrupadas em uma distância entre os pixels. A formulação apresentada utilizará a distância Euclidiana. [5]

1. Inicialize o centro de agrupamento K aleatoriamente;
2. atribua para cada x_i ao grupo do centro de agrupamento mais próximo, utilizando a distância euclidiana.

$$KM(X, C) = \sum_{i=1}^n \min \|x_i - c_j\|^2; j \in (1 \dots k)$$

3. Update cada centro e agrupamento c_k como a média de cada x_i pertencente a ele.
4. repita os passos 2 a 4 até os centros dos agrupamentos se tornarem estáveis.

4 Resultados

Foram utilizado na base da dados, dois extratores de características: o momento de HU e o LBP; esses extratores foram combinado com dois classificadores o perceptron simples e o kNN totalizando quatro resultados. O dataset foi dividido em duas partes, metade dos dados foram utilizados para treinamento e a outra metade foi utilizada para teste. Desta forma, podemos definir a taxa de acerto, que trata-se de uma razão entre o que foi predito e qual era a categoria verdadeira.

Utilizando o Momento de Hu como extrator de características temos que para um perceptron simples uma taxa de acerto de 57.1% enquanto utilizando o kNN com três vizinhos a

taxa de acerto aumentou para 78,8%. Aumentando a quantidade de vizinhos analisadas pelo algoritmo para 5 temos um aumento de quase um por cento, a taxa de acerto é de 79,7%. Porém, ao aumentar a quantidade de vizinhos para 7, temos um decrescimo na taxa de acerto, sendo abaixo de utilizarmos tanto 3 quanto 5 vizinhos, sendo de 77,8% . Já quando fizemos uma variação na quantidade de dados utilizados para teste, diminuindo para 30% a taxa de acerto foi de 79,5%.

A Matriz de confusão utilizando o kNN e o momento de Hu, para a condição com 3 vizinhos para o kNN e metade do dataset para treinamento e a outra metade para treino é apresentada a seguir, na qual a linha apresenta o valor real e a coluna o valor predito:

$$\begin{bmatrix} 62 & 0 & 1 & 0 & 0 & 3 & 2 & 0 & 5 & 4 \\ 0 & 99 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 64 & 12 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 6 & 78 & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 101 & 0 & 1 & 0 & 0 & 0 \\ 8 & 0 & 4 & 2 & 0 & 64 & 0 & 0 & 1 & 3 \\ 5 & 0 & 0 & 0 & 1 & 1 & 51 & 0 & 12 & 21 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 69 & 0 & 0 \\ 7 & 0 & 0 & 0 & 2 & 0 & 20 & 0 & 42 & 7 \\ 7 & 0 & 0 & 0 & 2 & 4 & 33 & 0 & 12 & 16 \end{bmatrix}$$

Já quando foi utilizado o extrator de características LBP a taxa de acerto para o classificador perceptron foi de 32,7% enquanto utiliza o kNN como classificador a taxa de acerto aumentou para 94,9% . Fazendo a mesma análise de variação da quantidade de vizinhos temos que para $k = 5$ sua taxa de acerto foi de 94.6% e para $k = 7$ a taxa de acerto aumentou para 95,7%. Porém, não teve variações significativas ao alterar a quantidade do dataset separado para o teste, sendo a taxa de acerto nessas condições de 95,2%. Tendo a seguinte matriz de confusão.

$$\begin{bmatrix} 77 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 99 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 73 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 69 & 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 103 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 74 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 89 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 68 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 77 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 71 \end{bmatrix}$$

5 Conclusão

O perceptron simples é a forma mais simples de redes neurais sendo utilizado para problemas linearmente separáveis e por causa dessas características de classificador, como o problema apresentado tinha atributos que não tinham essas características e por isso temos uma taxa de acerto bastante baixa.

Uma forma de melhoria para o resultados do perceptron simples seria a utilização de uma rede neural com uma camada oculta, desta forma, as descontinuidade e não linearidades dos atributos poderiam ser tratadas por esse tipo de classificador.

Já o kNN que utiliza a distância entre os vizinhos teve uma taxa melhor de acerto, mesmo utilizando um número pequeno de vizinhos já temos uma taxa de acerto maior que 90%. Porém, a alteração da quantidade de vizinhos a serem analisadas não trouxe uma melhoria significativa, vale ressaltar que ao aumentar a quantidade de vizinhos aumenta também a complexidade temporal e espacial do algoritmo, uma vez que será calculado a distância para mais pontos.

Vale ressaltar que as taxas de acerto pode variar cada vez que o algoritmo for executado, visto que ambos algoritmos de classificação são influenciados pelos dados escolhidos para treinamento e que no caso que o tempo de convergência do algoritmo depende da escolha dos bias para o perceptron simples e a escolha dos primeiros centros para o kNN.

A escolha do extrator de características também teve uma influência nos resultados obtidos, vimos que o momento de Hu, teve resultados inferiores ao LBP. Para trabalhos futuros podemos testar as influências de cada atributo obtido dos extratores nos resultados obtidos pelos classificadores. Além, disso pode-se combinar outras técnicas de extração de características com outros algoritmos de classificação para obter melhores resultados.

Referências

- [1] Opencv library. <https://opencv.org/>. Accessed: 2019-03-12.
- [2] scikit-learn: machine learning in python — scikit-learn 0.21.1 documentation, how-published = <https://scikit-learn.org/stable/modules/classes.html>, note = Accessed: 2019-03-20.
- [3] C. Solomon and T. Breckon. *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab*. Wiley, 2011.
- [4] R.C. Gonzalez and R.E. Woods. *Processamento Digital De Imagens*. ADDISON WESLEY BRA, 2008.
- [5] J. Gama, K. Faceli, A.C. Lorena, and A.C.P.L.F. De Carvalho. *Inteligência artificial: uma abordagem de aprendizado de máquina*. Grupo Gen - LTC, 2011.
- [6] S.J. Russell, S.J. Russell, P. Norvig, and E. Davis. *Artificial Intelligence: A Modern Approach*. Prentice Hall series in artificial intelligence. Prentice Hall, 2010.