

RECONHECIMENTO DE CARACTERES UTILIZANDO LÓGICA NEBULOSA NA CLASSIFICAÇÃO E SELEÇÃO DE PADRÕES COM AUTO-APRENDIZADO

Jomar Fernandes Nascimento Francisco José Gomes

Faculdade de Engenharia da UFJF
Dept. Energia Elétrica
CP 422 - Fone (032)229-3424
CEP 36100-970 - Juiz de Fora - MG

RESUMO

Apresenta-se neste trabalho, um novo método para reconhecimento de caracteres utilizando Lógica Nebulosa ("Fuzzy Logic"), Distância de Levenshtein, algoritmos para redução de bitmaps e técnicas de auto-aprendizado. O método não se restringe apenas ao reconhecimento de letras, sendo possível reconhecer também dígitos, símbolos, caracteres chineses, kanji, uma vez que ele se detém em analisar a morfologia do mesmo. A partir deste novo método foi implementado um ambiente computacional integrado para sua utilização. Os resultados obtidos foram satisfatórios, comprovando aplicabilidade da lógica nebulosa na classificação e seleção de padrões. O auto-aprendizado se mostrou de grande importância nos processos de treinamento do sistema e posterior utilização do mesmo através da redução a níveis muito baixos da taxa de erro, sem comprometer a velocidade de reconhecimento.

Palavras chaves: Reconhecimento Ótico de Caracteres, Lógica Nebulosa, Distância de Levenshtein.

CHARACTER RECOGNITION USING FUZZY LOGIC ON THE CLASSIFICATION AND SELECTION OF PATTERNS WITH AUTO LEARNING

ABSTRACT

This work presents a method for character recognition using Fuzzy Logic, Levenshtein Distance, algorithms for bitmap reduction and auto-learn techniques. This method is not restricted for recognition of letters, it is also possible to recognize numerical characters, symbols, Chinese characters, kanji, because it analyses the character's morphology. Using this new method, an integrated computational environment was developed to use it. The results were very good, making sure that Fuzzy Logic can be used in applications where pattern classification and selection are needed. The auto-learn process shows itself very important at the training process and forward utilization of the system decreasing the error tax to very short values, without changing the recognition speed.

Key words: Character recognition, Fuzzy Logic, Levenshtein Distance.

1 INTRODUÇÃO

O reconhecimento ótico de caracteres tem recebido considerável atenção nos últimos anos devido a crescente dependência do computador para processamento de informações.

Vários métodos foram desenvolvidos e são hoje largamente utilizados em aplicações OCR ("Optical Character Recognition"). Destacam-se os métodos de análise estatística (Al-Yousefi, 1992), redes neuronais (Knerr, 1992) entre outros.

A avaliação de cada método é realizada em geral através de dois parâmetros: Taxa de acerto e a Velocidade de reconhecimento, parâmetros estes que qualificam e definem sua área de aplicação.

O objetivo deste trabalho consiste no desenvolvimento de um novo método de reconhecimento, que possua uma relação Taxa de Acerto x Velocidade de Reconhecimento tão boa quanto os métodos já conhecidos. Para tal, foram utilizadas várias tecnologias que juntas apresentam o resultado esperado.

O reconhecimento é feito a partir da análise do caracter e obtenção de suas características principais, ligadas à sua morfologia

e posteriormente através da comparação com informações sobre outros caracteres previamente armazenados em uma base de dados, escolhendo o que mais se assemelha.

Este processo de seleção tende a ser bastante subjetivo, haja visto que as características dificilmente se assemelham em 100%. Torna-se assim necessária uma forma de classificar esta semelhança em grupos, e dentro destes grupos seu grau de pertinência.

Isto é conseguido através da utilização da distância de Levenshtein na classificação e de Lógica Nebulosa (Fuzzy) no processo de avaliação e determinação.

Em caso de erro na seleção ou não localização de um caso ótimo, o sistema possibilita ao usuário corrigir e/ou incluir a nova informação na base de dados, aumentando progressivamente a taxa de acerto.

O trabalho está dividido da seguinte forma: O capítulo 2 apresenta as tecnologias e metodologias utilizadas, o capítulo 3 descreve os algoritmos desenvolvidos. Os resultados e conclusões são apresentados respectivamente nos capítulos 4 e 5.

2 METODOLOGIA

2.1 LÓGICA NEBULOSA

A lógica Fuzzy foi introduzida pelo professor Lotfi A. Zadeh, professor da Universidade da Califórnia em 1965 (Zadeh, 1965), e surgiu da necessidade de modelar sistemas em que se desejava a habilidade de definir conclusões e gerar respostas baseadas em informação vaga, ambígua, qualitativa, incompleta ou imprecisa.

A figura 1 mostra o fluxo de dados em um sistema Fuzzy.

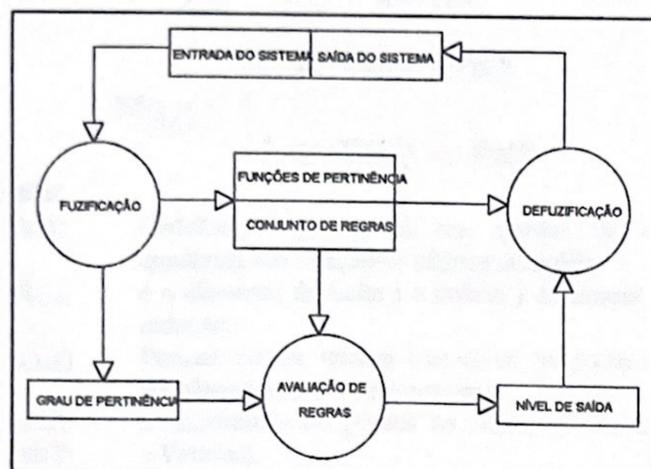


Fig. 1 - Fluxo de Dados num Sistema Modelado através de Lógica Nebulosa

2.2 DISTÂNCIA DE LEVENSHTEIN

Existem várias aplicações onde é necessário comparar seqüências de valores, e obter uma medida que defina o grau de diferenciação entre as mesmas; entre elas destacam-se reconhecimento de voz, análise geológica, análise de impressões digitais, biologia molecular entre outras (Valdés, 1992). Boa parte dos algoritmos hoje utilizados derivam do trabalho realizado por V.I. Levenshtein (Levenshtein, 1965).

Pode-se definir a distância de Levenshtein entre duas seqüências através da seguinte relação recursiva:

$$D(a_i, b_j) = \min \begin{cases} D(a_{i-1}, b_j) + w(a[i], \phi) \\ D(a_{i-1}, b_{j-1}) + w(a[i], b[j]) \\ D(a_i, b_{j-1}) + w(\phi, b[j]) \end{cases}$$

onde:

- $D(a_i, b_j)$ é a distância de uma seqüência de tamanho i para uma seqüência b de tamanho j ;
- $w(a[i], \phi)$ é o peso de eliminar o elemento $a[i]$;
- $w(a[i], b[j])$ é o peso de substituir o elemento $a[i]$ pelo elemento $b[j]$;
- $w(\phi, b[j])$ é o peso de inserir o elemento $b[j]$.

3 ALGORÍTMOS UTILIZADOS

3.1 VISÃO GERAL DO MÉTODO

A Fig. 2 apresenta um fluxograma completo do método incluindo ainda procedimentos utilizados na implementação do ambiente computacional de reconhecimento.

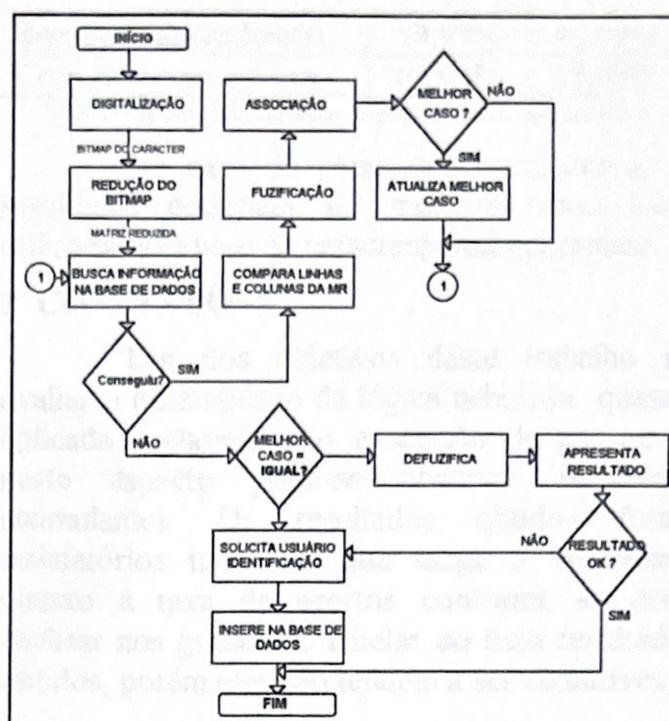


Fig. 2 - Fluxograma do Sistema

3.2 FUNÇÕES DE PERTINÊNCIA E CONJUNTO DE REGRAS FUZZY

A lógica Fuzzy foi utilizada para viabilizar a comparação de dois caracteres, utilizando características de subjetividade inerentes ao raciocínio humano.

A entrada do sistema é a distância de Levenshtein entre cada linha e cada coluna das matrizes reduzidas dos caracteres em análise e sua função de pertinência pode ser observado na Fig. 3.

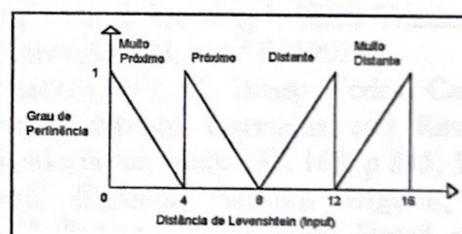


Fig. 3 - Funções de Pertinência para Entrada no Sistema (Fuzificação)

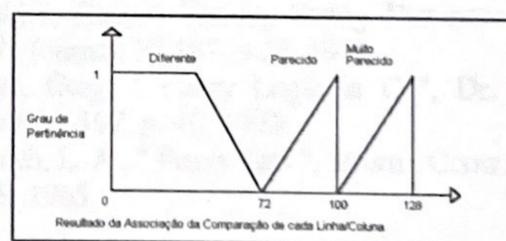


Fig. 4 - Funções de Pertinência para Saída do Sistema (Defuzificação)

Para definição das funções de pertinência apresentadas e conjunto de regras, foi utilizada modelagem heurística, baseada em testes e resultados obtidos do sistema para cada conjunto de funções experimentado.

Linha / Linha +1	Muito Distante	Distante	Próximo	Muito Próximo
Muito Distante	D	D	D	D
Distante	D	D	D	P
Próximo	D	D	P	P
Muito Próximo	D	P	P	I

Tab. 1 - Conjunto de Regras para Associação da comparação de Linha e Colunas

D=Diferente, P = Parecido, I = Igual

3.3 REDUÇÃO DO MAPA DE BITS SEM PERDA DE INFORMAÇÃO

Para solucionar o problema da redundância de pontos e do ruído da imagem, foi desenvolvido um algoritmo capaz de transformar imagens de qualquer tamanho em uma de 16x16 pontos, dispostos na forma de uma matriz chamada de matriz de redução.

O termo geral da matriz de redução pode ser expresso como:

$$U(i,j) = \frac{\sum_{y=i*SliceY}^{(i+1)*SliceY} \sum_{x=j*SliceX}^{(j+1)*SliceX} \varphi(x,y)}{\sum_0^{LenY} \sum_0^{LenX} \varphi(x,y)} \cdot 100\%$$

$$M(i,j) = \begin{cases} 0, & \text{se } U(i,j) > FatF \\ 1, & \text{se } U(i,j) \leq FatF \end{cases}$$

onde:

- U(i,j) Distribuição percentual dos pontos de um quadrante em relação ao bitmap completo;
- M_(i,j) é o elemento da linha i e coluna j da matriz de redução;
- φ(x,y) Função binária, retorna 1 se existe um ponto nas coordenadas x,y, 0 caso contrário
- LenX Comprimento em pontos do bitmap (Horizontal e Vertical)
- LenY Comprimento em pontos do bitmap (Horizontal e Vertical)
- SliceX é o tamanho em pontos das regiões horizontal e vertical do bitmap analisado
- SliceY é o tamanho em pontos das regiões horizontal e vertical do bitmap analisado
- FatF Fator de Filtro.

As Figs. 5 mostra exemplo de redução do bitmap correspondente à letra A manuscrita.

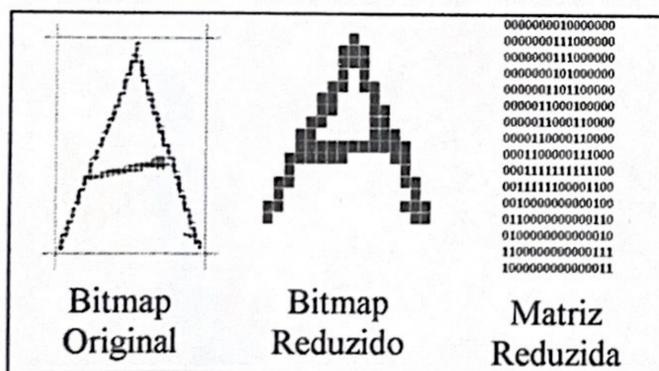


Fig. 6 - Exemplo de redução de bitmap da Letra A Maiúscula Manuscrita

3.4 AUTO-APRENDIZADO

Para possibilitar um aumento na taxa de acertos deste método tornou-se necessária a inclusão de um módulo de auto-aprendizado e correção, onde a cada erro ou a cada caracter não reconhecido, torna possível uma correção.

4 RESULTADOS OBTIDOS

Foram realizados testes com caracteres impressos e manuscritos, utilizando e não utilizando o auto-aprendizado.

Para treinamento inicial do sistema foi utilizado um único elemento de cada tipo de caracter pertencente ao conjunto de teste.

Teste	Taxa de Acertos	Taxa de Erros
Sem auto-aprendizado	88,23%	11,77%
Com auto-aprendizado	95,30%	4,70%

Tab. 2 - Resultado do teste com caracteres impressos

Teste	Taxa de Acertos	Taxa de Erros
Sem auto-aprendizado	58,97%	41,03%
Com auto-aprendizado	65,04%	34,96%

Teste	Taxa de Acertos	Taxa de Erros
Sem auto-aprendizado	58,97%	41,03%
Com auto-aprendizado	65,04%	34,96%

Tab. 3 - Resultado do teste com caracteres manuscritos

No caso de caracteres manuscritos, os resultados poderiam ser melhores caso fosse utilizada uma base de caracteres mais completa.

5 CONCLUSÕES

Um dos objetivos deste trabalho foi avaliar o desempenho da lógica nebulosa quando aplicada a classificação e seleção de padrões e neste aspecto pode-se observar resultados estimulantes. Os resultados obtidos foram satisfatórios tanto no que tange à velocidade quanto à taxa de acertos conforme se pode analisar nos gráficos e tabelas do item resultados obtidos, porém eles não tendem a ser definitivos.

6 BIBLIOGRAFIA

- 1) Al-Yousefi, H.; S.S. Udpa, " Recognition of Arabic Characters ", Transactions on Systems, Man and Cybernetics, vol. 14, Nº 8, 1992
- 2) Avitzur, Ron, " Your Own Handprinting Recognition Engine ", Dr. Dobb's Journal, Nº 187, p. 32, 1992.
- 3) Filho, Ogê Marques, " Reconhecimento de Caracteres Utilizando Correlação com Máscaras Binárias ", 8º Congresso Brasileiro de Automática, 1990
- 4) Knerr, Stefan; Léon Personnaz; Gérard Dreyfus, " Handwritten Digit Recognition by Neural Networks with Single-Layer Training ", IEEE Transactions on Neural Networks, vol. 3, Nº 6, 1992
- 5) Levenshtein, V.I., " Binay Codes Capable of Correcting Deletions, Insertions and Reversals ", Doklady Akadaemii Nauk SSR 163, p 845, 165.
- 6) Tamura, Shinichi; Seihaku Higuchi; Kokichi Tanaka, " Pattern Classification Based on Fuzzy Relations ", Transactions on Systems, Man and Cybernetics, vol 1, Nº 1, 1971
- 7) Valdés, Ray, " Finding String Distances ", Dr. Dobb's Journal, Nº 187, p.56, 1992.
- 8) Viot, Greg, " Fuzzy Logic in C ", Dr. Dobb's Journal, Nº 197, p. 40, 1993.
- 9) Zadeh, L. A., " Fuzzy Sets ", Inform. Contr., vol. 8, p. 338, 1965.

AGRADECIMENTOS A CAPES PELO APOIO FINANCEIRO