Algoritmo de Wang-Mendel para aprendizagem de regras fuzzy

Renato Lopes Moura

16 de dezembro de 2016

Sumário

Visão Geral

Esta apresentação descreve o método geral proposto por Li-Xin Wang e Jerry Mendel [**?**] para geração de uma base de regras fuzzy a partir de dados numéricos.

O método consiste de 5 passos, que serão detalhados a seguir. E após, serão apresentados alguns exemplos de aplicação em problemas diversos.

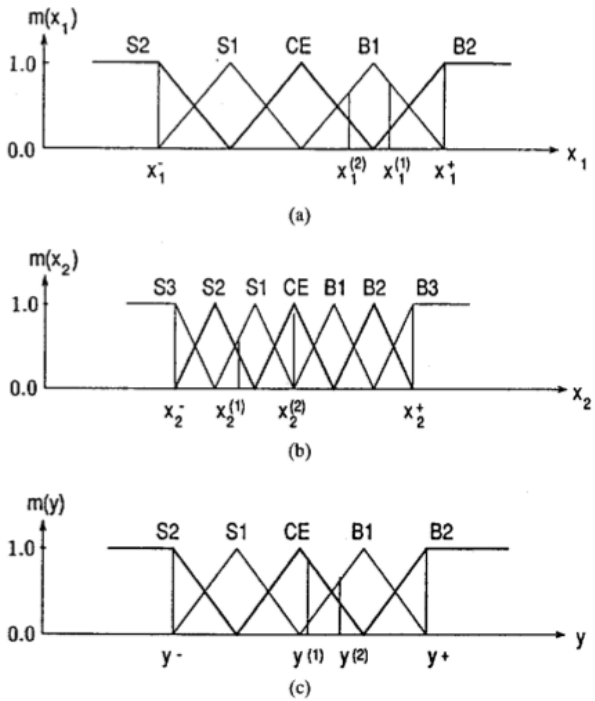
Para facilitar o entendimento do método vamos considerar um caso simples com **2 variáveis de entrada** e **1 variável de saída**, que poderá ser facilmente estendido para casos mais gerais ao final.

Passo 1: Divisão dos espaços de entradas e saídas

Inicialmente, é necessário dividir os intervalos de valores possíveis das variáveis de entrada e saída em regiões fuzzy.
Supondo que os domínios das variáveis de entrada e saída sejam respectivamente:

$$[x_1^-, x_1^+], [x_2^-, x_2^+], [y^-, y^+]$$

Podemos criar as seguintes regiões fuzzy utilizando funções de pertinência triangulares:



Passo 2: Geração de regras fuzzy a partir dos dados

Em seguida, para cada par **entradas-saída** são determinados os graus de pertinência e é gerada uma regra. Por exemplo considerando o seguinte par:

$$(x_1^{(n)}, x_2^{(n)}; y^{(n)})$$

Podemos gerar a seguinte regra:

$$(x_1^{(n)}, x_2^{(n)}; y^{(n)}) \Rightarrow [(x_1^{(n)}(0.8 \text{ em } B1), x_2^{(n)}(0.7 \text{ em } S1); y^{(n)}(0.9 \text{ em } CE)] \Rightarrow \text{Regra n}$$

SE x_1 é $B1$ e x_2 é $S1$, **ENTÃO** y é CE

Passo 3: Associando um grau a cada regra

Após o passo anterior devemos ter uma grande quantidade de regras (1 para cada par entradas-saída), e inclusive regras conflitantes entre si (antecedentes iguais mas consequentes diferentes).
Para resolver tais conflitos, é associado um "grau de confiabilidade"a cada regra dado por:

$$D(\text{Regra n}) = m_{B1}(x_1^{(n)}).m_{S1}(x_2^{(n)}).m_{CE}(y^{(n)})$$

E então nos casos de conflito são descartadas as regras com menor "grau de confiabilidade".

Nesta etapa também é possível agregar o conhecimento de um especialista e/ou controlar a influência de dados que possam conter erros de medição. Isto é feito acrescentando-se um novo parâmetro $m^{(n)}$ à equação anterior:

$$D(\text{Regra n}) = m_{B1}(x_1^{(n)}).m_{S1}(x_2^{(n)}).m_{CE}(y^{(n)}).m^{(n)}$$

Passo 4: Criação de uma Base de Regras

No caso simples do nosso exemplo são possíveis 35 regras distintas conforme a tabela a seguir:

		x2						
		S3	S2	S1	CE	B1	B2	B3
x1	S2							
	S1							
	CE							
	B1							
	B2							

Tabela: Base de regras

Passo 4: Criação de uma Base de Regras

Idealmente, cada entrada da tabela anterior deve ter um valor associado que foi obtido no passo 3. Porém, podem existir combinações de antecedentes que não foram cobertas pelo conjunto de treinamento.

Nestes casos, podem ser criadas regras linguísticas com o apoio de um especialista para completar a base de regras.
Este artifício também pode ser utilizado para substituir regras que apresentem um baixo "grau de confiabilidade".

Passo 5: Inferência e Defuzzificação

Por fim, com as devidas adaptações, podemos utilizar diferentes métodos de inferência e de defuzzificação dos resultados para tratar dados novos.

Os métodos de inferência e defuzzificação mais utilizados são Mamdani e Centróide, respectivamente.

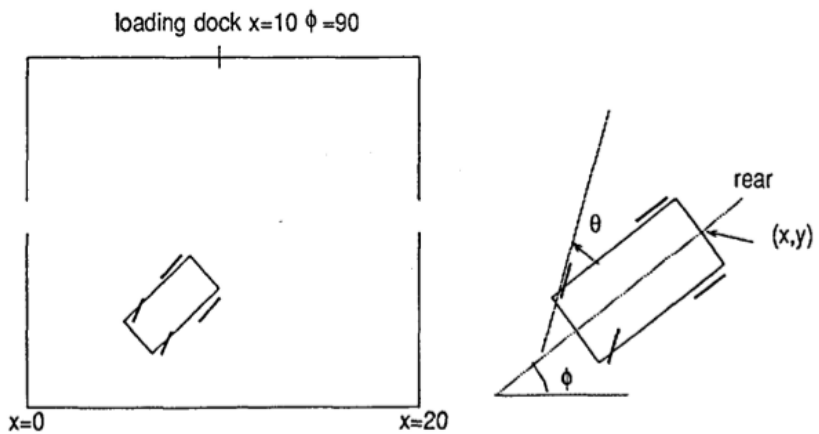
Aplicação 1: Back-Upper Truck

Trata-se de um problema clássico da literatura que consiste em estacionar um caminhão de ré em uma doca.
As variáveis de entrada são:

- x - posição do caminhão no eixo horizontal
- ϕ - ângulo do caminhão em relação ao eixo horizontal

E a variável de saída é:

- θ - ângulo das rodas dianteiras em relação ao eixo do caminhão



Aplicação 1: Back-Upper Truck

Para a obtenção do conjunto de dados de treinamento foram utilizados 14 estados iniciais (posição e ângulo) e as seguintes equações cinemáticas aproximadas:

$$x(t+1) = x(t) + \cos[\phi(t) + \theta(t)] + \sin[\theta(t)]\sin[\phi(t)]$$

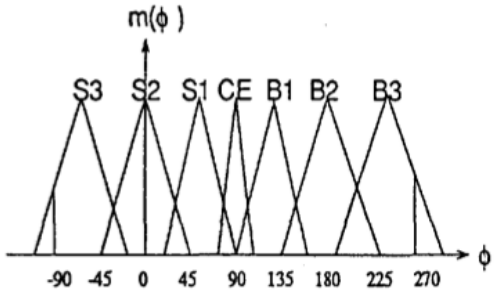
$$\phi(t+1) = \phi(t) - \sin^{-1}[\frac{2\sin(\theta(t))}{b}]$$

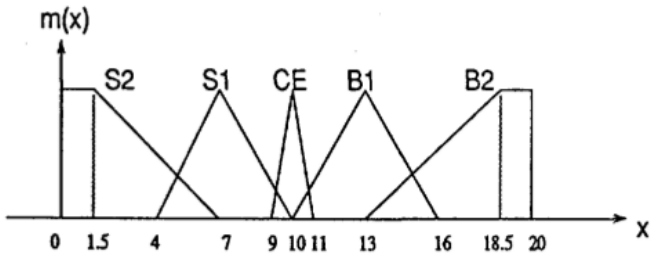
onde b é o tamanho do caminhão, no nosso caso $b = 4$.

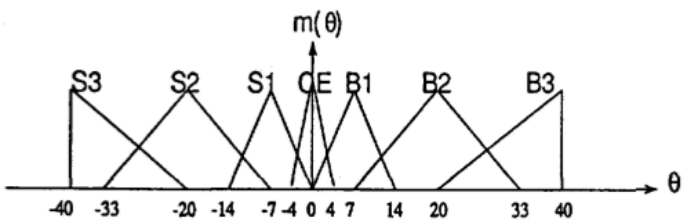
Os estados iniciais e dados de treinamento podem ser vistos em [**?**]

Aplicação 1: Back-Upper Truck

Os conjuntos fuzzy definidos para as variáveis de entrada e saída estão representados nas figuras a seguir:







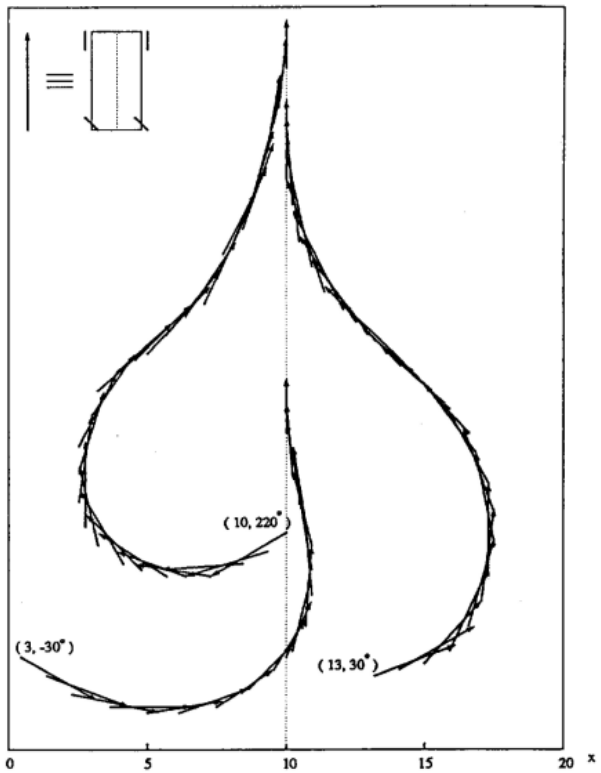
Aplicação 1: Back-Upper Truck

E a base de regras obtida a partir dos dados numéricos foi a seguinte:

ϕ \ x	S2	S1	CE	B1	B2
S3	S2	S3			
S2	S2	S3	S3	S3	
S1	B1	S1	S2	S3	S2
CE	B2	B2	CE	S2	S2
B1	B2	B3	B2	B1	S1
B2		B3	B3	B3	B2
B3				B3	B2

Aplicação 1: Back-Upper Truck

Os resultados obtidos foram:



Aplicação 2: Série temporal de Mackey-Glass

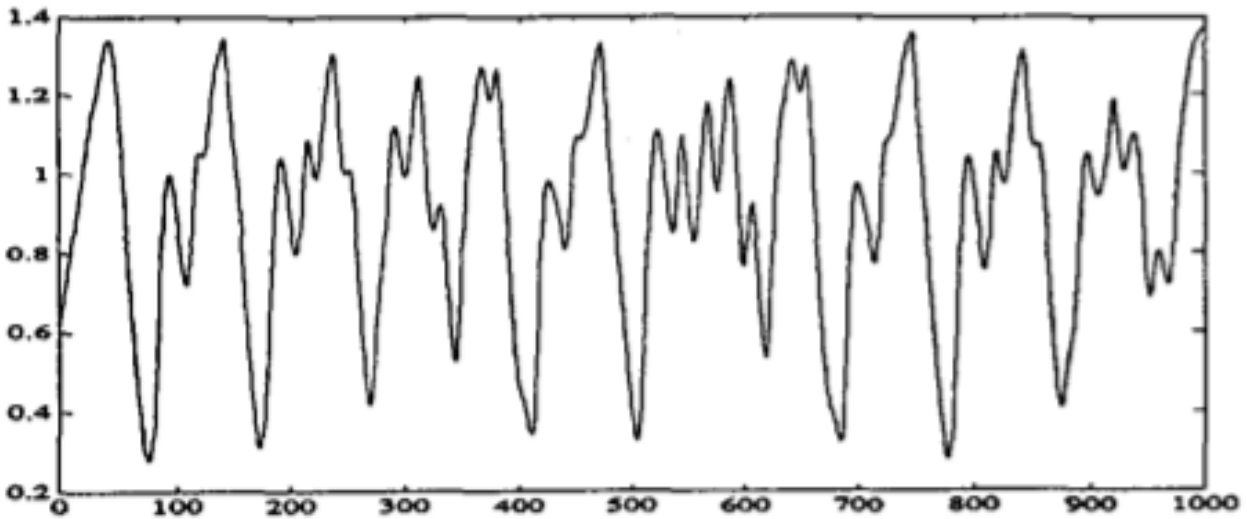
A série temporal de Mackey-Glass é gerada a partir da seguinte equação diferencial com atraso:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t)$$

Quando $\tau > 17$ a série apresenta um comportamento caótico, ou seja, se torna uma série temporal aparentemente randômica.

Aplicação 2: Série temporal de Mackey-Glass

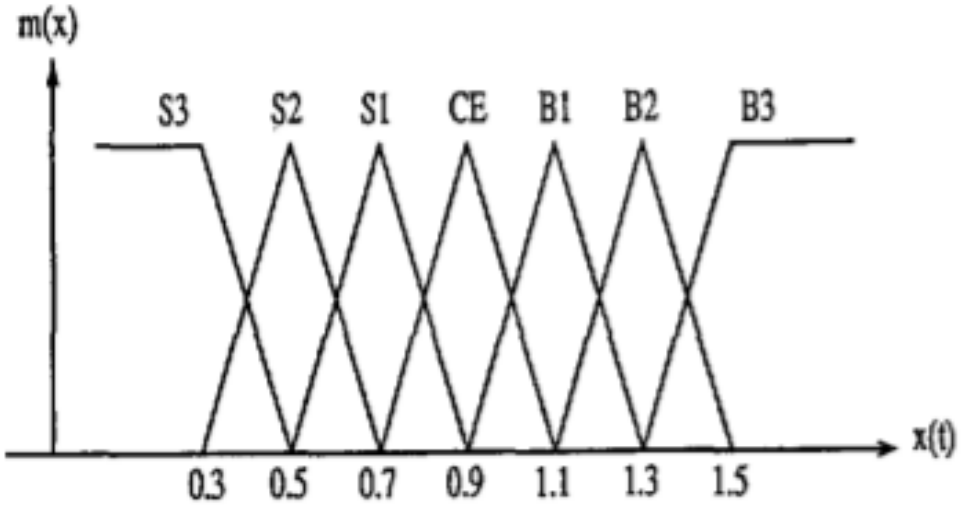
Utilizando $\tau = 30$ e calculando os primeiros 1000 pontos da série, obtemos o seguinte gráfico:



Aplicação 2: Série temporal de Mackey-Glass

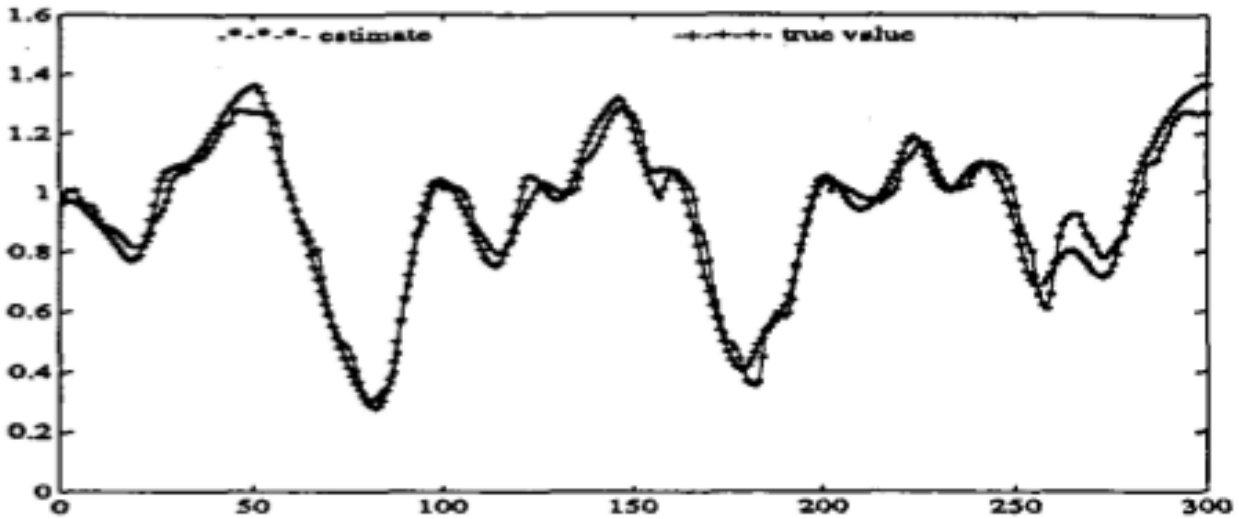
Vamos utilizar 9 pontos consecutivos para prever o valor do próximo ponto.

Neste caso, os espaços das variáveis de entrada e de saída serão os mesmos, e podem ser divididos da seguinte maneira:



Aplicação 2: Série temporal de Mackey-Glass

Utilizando os primeiros 700 pontos da série como conjunto de treinamento e os últimos 300 como conjunto de testes, obtemos o resultado da figura:



Referências

L. Wang and J. M. Mendel, *Generating fuzzy rules by learning from examples* IEEE Trans. Syst., Man, Cybern., vol. 22, no. 6, pp. 1414-1427, (1992).

L. Wang and J. M. Mendel, *Generating fuzzy rules from numerical data, with applications*, USC SIPI Rep. No. 169, Univ. Southern Calif., Los Angeles, (1991).