

本文介绍了 TS 型模糊系统，由 Takagi 和 Sugeno 两位学者在 1985 年提出，主要思想是将非线性系统用许多线段相近的表示出来，即将复杂的非线性问题转化为在不同小线段上的问题。

- 1. TS 模糊系统
 - 1.1. 推理过程
 - 1.2. 特性
 - 1.3. 辨识算法
- 2. TS 模糊控制
- 3. 广义 TS 模糊系统
- 4. 应用
 - 4.1. Fuzzy Control
 - 4.2. Fuzzy Neural Network
 - 4.2.1. 网络结构
 - 4.2.2. 网络参数辨识
 - 4.3. Trajectory Prediction
- 5. 参考文献

1. TS 模糊系统

Tomohiro Takagi and Michio Sugeno. **Fuzzy Identification of Systems and Its Applications to Modeling and Control**[J]. Fuzzy Identification of Systems, 1993.

A mathematical tool to build a fuzzy model of a system where fuzzy implications and reasoning are used is presented in this paper. The premise of an implication is the description of fuzzy subspace of inputs and its consequence is a linear input-output relation. The method of identification of a system using its input-output data is then shown. Two applications of the method to industrial processes are also discussed: a water cleaning process and a converter in a steel-making process.

TS 模糊模型是由多个线性系统对同一个非线性系统进行拟合，利用模糊算法进行输入变量的解构，通过模糊演算推理再去模糊化，生成数条代表每组输入与输出关系的方程。

假设模糊集为 A ，隶属度函数为 $A(x)$ ， x 属于某论域 X 。“ x 属于 A 且 y 属于 B ”表达为

$$|x \text{ is } A \text{ and } y \text{ is } B| = A(x) \wedge B(y)$$

对于离散系统模型，令 R_i 表示模糊系统的第 i 条规则，其一阶 TS 模糊系统典型的模糊蕴含条件 (Implication) 句为

$$R_i : \quad \text{if } f(x_1 \text{ is } A_1, \cdots, x_k \text{ is } A_k) \quad \text{then } y = g(x_1, \cdots, x_k)$$

在实际应用中, f 为 *and* 连接符, g 为线性函数, 即

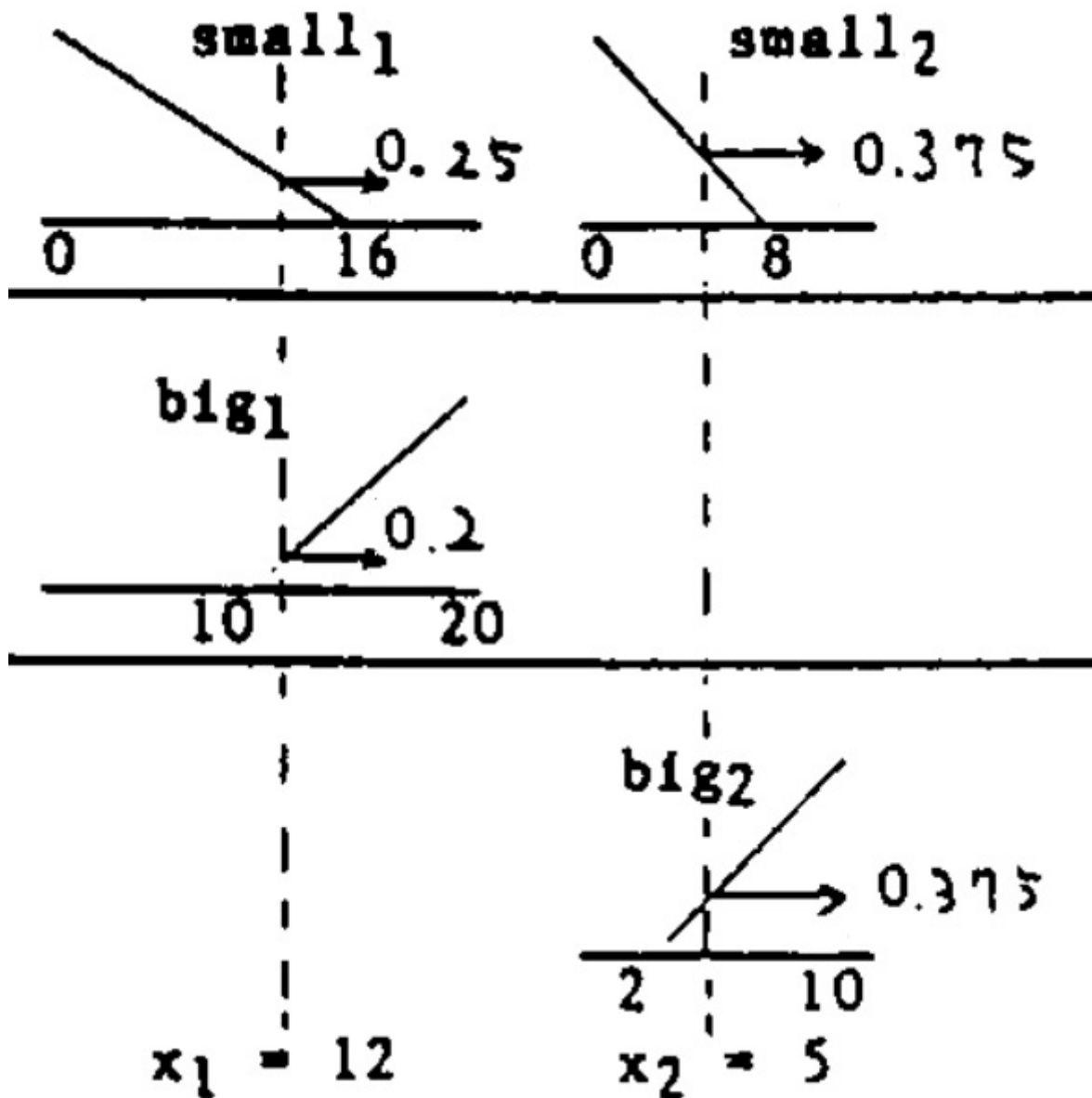
$$R : \quad \text{if } x_1 \text{ is } A_1 \text{ and } \cdots \text{ and } x_k \text{ is } A_k \quad \text{then } y = p_0 + p_1 x_1 + \cdots + p_k x_k)$$

1.1. 推理过程

假设有 3 个上述格式的蕴含条件 $R_i, i = 1, \cdots, 3$, 分别为

$$\begin{array}{ll} R_1 : & \text{if } x_1 \text{ is } small_1 \text{ and } x_2 \text{ is } small_2 \quad \text{then } y = x_1 + x_2 \\ R_2 : & \text{if } x_1 \text{ is } big_1 \quad \text{then } y = 2x_1 \\ R_3 : & \text{if } x_2 \text{ is } big_2 \quad \text{then } y = 3x_2 \end{array}$$

前提 (Premise) 中涉及到的隶属度函数定义为



假设输入 $x_1 = 12, x_2 = 5$, 那么三个前提下的结论 (Consequence) 为

$$y_1 = x_1 + x_2 = 17$$

$$y_2 = 2x_1 = 24$$

$$y_3 = 3x_2 = 15$$

相应的三个真值 (True Value) 为

$$t_1 = \text{small}_1(x_1) \wedge \text{small}_2(x_2) = 0.25 \wedge 0.375 = 0.25$$

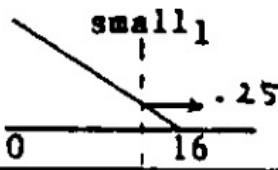
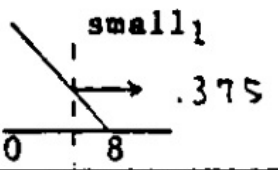
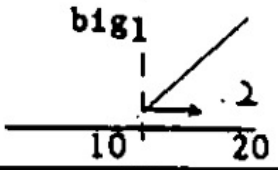
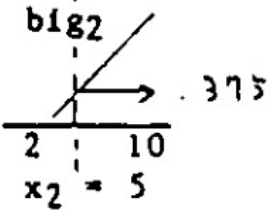
$$t_2 = \text{big}_1(x_1) = 0.2$$

$$t_3 = \text{big}_2(x_2) = 0.375$$

那么最终 y 的取值为 (此处采用加权平均法)

$$y = \frac{t_1 y_1 + t_2 y_2 + t_3 y_3}{t_1 + t_2 + t_3} \approx 17.8$$

用一张表格可以列写如下

Implication	Premise	Consequence	Tv	
R1			$y = 12 + 5$ $= 17$	$.25 \wedge .375$ $= .25$
R2			$y = 2 \times 12$ $= 24$.2
R3	$x_1 = 12$	 $x_2 = 5$	$y = 3 \times 5$ $= 15$.375

1.2. 特性

优点：

- 相比分段线性逼近，数学形式更紧凑，连接处比较平滑；
- 相比原始的非线性函数，更加简明，方便进一步处理；
- 模糊划分可以包含有意义的语义条件，方便的讲人类语言规则形式表达的先验知识融入到模型建立过程中（模糊逻辑的功效）；
- 万能逼近定律表明 TS 模糊系统能以任意精度逼近非线性模型，适用于广泛类型的非线性系统。

另一方面，TS 模糊系统存在以下问题

- 隶属度函数全部由直线组成，不具备自适应性
- 不能保证参数的最优性
- 模糊规则数目无法最佳确定，即无法预知模型的复杂程度

1.3. 辨识算法

需要确定以下三个部分

- x_i, \dots, x_k , 前提变量;
- A_1, \dots, A_k , 隶属度函数的参数, 简记为隶属度参数;
- p_0, p_1, \dots, p_k , 结论中的参数。

注意, 前提中的变量不需要全部出现。前两个部分的确定和变量如何划分到模糊子空间有关, 最后一个部分与模糊子空间中如何描述输入输出关系有关。论文作者提出依次逐层考虑如何确定。

假设一个一般的系统表示如下

$$\begin{aligned}
 R_1 : \quad & \text{if } x_1 \text{ is } A_1^1, \dots, x_k \text{ is } A_k^1 \\
 & \text{then } y = p_0^1 + p_1^1 \cdot x_1 + \dots + p_k^1 \cdot x_k \\
 & \vdots \\
 R_n : \quad & \text{if } x_1 \text{ is } A_1^n, \dots, x_k \text{ is } A_k^n \\
 & \text{then } y = p_0^n + p_1^n \cdot x_1 + \dots + p_k^n \cdot x_k
 \end{aligned}$$

那么输出为

$$y = \frac{\sum_{i=1}^n (A_1^i(x_1) \wedge \dots \wedge A_k^i(x_k)) \cdot (p_0^i + p_1^i x_1 + \dots + p_k^i x_k)}{\sum_{i=1}^n (A_1^i(x_1) \wedge \dots \wedge A_k^i(x_k))}$$

假设

$$\beta_i = \frac{A_1^i(x_1) \wedge \dots \wedge A_k^i(x_k)}{\sum_{i=1}^n (A_1^i(x_1) \wedge \dots \wedge A_k^i(x_k))}$$

那么

$$y = \sum_{i=1}^n \beta_i (p_0^i + p_1^i x_1 + \dots + p_k^i x_k)$$

当给定一组输入输出数据 $x_{1j}, \dots, x_{kj} \rightarrow y_j$ ($j = 1, \dots, m$) 时, 可以通过 least squares method 来确定参数 $p_0^i, p_1^i, \dots, p_k^i$ 。

2. TS 模糊控制

T. Taniguchi; K. Tanaka; H. Ohtake; H.O. Wang. **Model construction, rule reduction, and robust compensation for generalized form of Takagi-Sugeno fuzzy systems**. IEEE Transactions on Fuzzy Systems (Volume: 9, Issue: 4, Aug 2001).

在线性矩阵不等式 (linear matrix inequality, LMI) 设计框架下, 基于 TS 模糊模型的非线性控制得以广泛应用。一般分为三个阶段:

- 第一阶段: 对非线性被控对象的模糊建模
 - 利用输入输出数据进行模糊模型辨识
 - 或 基于分区非线性思想的模糊系统构建 (模糊 IF-THEN 规则)
- 第二阶段: 模糊控制规则推导, 它反映了模糊模型的规则结构, 它通过所谓的并行分布式补偿 (PDC) 实现
- 第三阶段: 模糊控制器设计, 即确定反馈增益。

This paper presents a systematic procedure of fuzzy control system design that consists of fuzzy model construction, rule reduction, and robust compensation for nonlinear systems.

本文提出了一种模糊控制系统设计的系统程序, 该程序由模糊模型构建, 规则约简和非线性系统的鲁棒补偿组成。

3. 广义 TS 模糊系统

将 TS 模糊系统进行规范化描述如下。

给定 m 个输入向量 x_1, \dots, x_m , n 条模糊规则为 R_1, \dots, R_n , 第 i 条模糊规则的模糊子集分别为 A_1^i, \dots, A_m^i (相应的隶属度函数为 $A_j^i(x_j)$), 各个模糊规则的真值为 G_1, \dots, G_n , 各个模糊规则对应的结论为 y_1, \dots, y_n , 最终输出为 y , 那么采用加权平均法的 TS 模糊系统为

$$y = \frac{\sum_{i=1}^n G_i y_i}{\sum_{i=1}^n G_i}$$

$$G_i = \prod_{j=1}^m A_j^i(x_j)$$

其中 \prod 为模糊化算子, 通常采取**取小** " \wedge " 或者 **代数积** " \cdot " 计算。

若隶属度函数采用高斯隶属度函数形式, 则可得到具有 m 输入单输出、模糊规则数为 n 的广义 TS 模糊系统

$$y = \frac{\sum_{i=1}^n G_i y_i}{\sum_{i=1}^n G_i}$$

$$G_i = \prod_{j=1}^m A_j^i(x_j) = \prod_{j=1}^m \exp\left(-\left|\frac{x_j - b_j^i}{a_j^i}\right|\right)$$

广义 TS 模糊系统可以以任意精度逼近被控对象，而模型的参数可以通过参数辨识方法获得。

4. 应用

4.1. Fuzzy Control

Robust L_1 Observer-Based Non-PDC Controller Design for Persistent Bounded Disturbed TS Fuzzy Systems

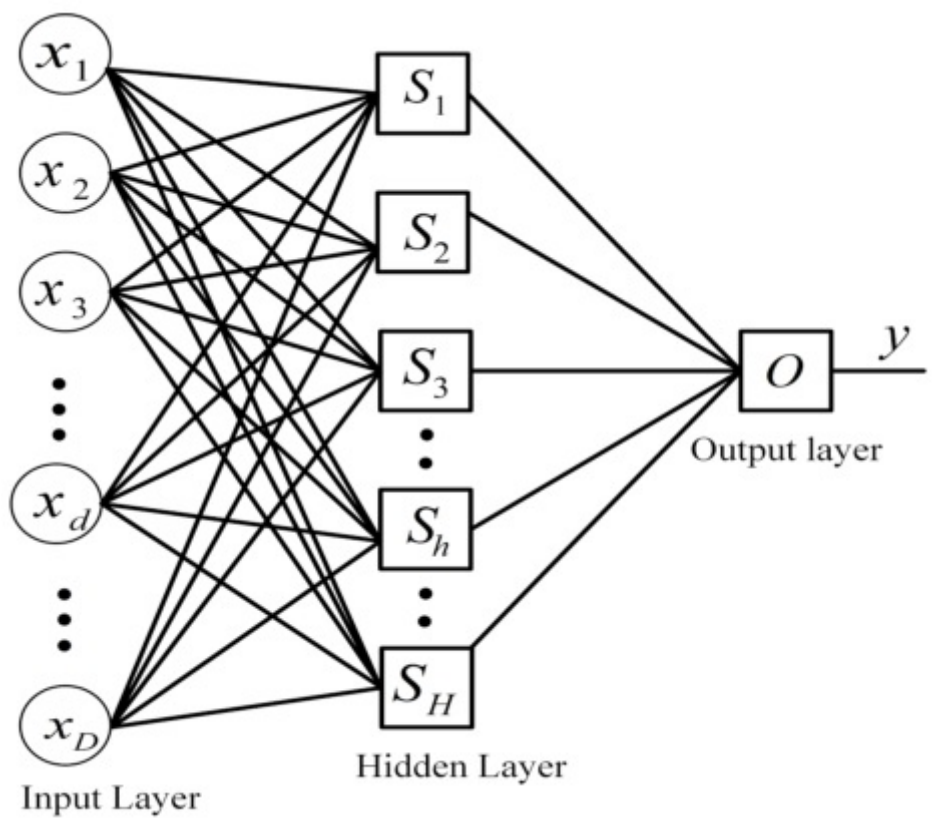
4.2. Fuzzy Neural Network

2017 . Developing deep fuzzy network with Takagi Sugeno fuzzy inference system. IEEE Transactions on Fuzzy System

4.2.1. 网络结构

提出了一种新型的三层 TS Deep Fuzzy Network (TSDFN) 网络架构。

TSDFN 的网络架构如下图所示



图中，隐层（hidden layer）中的每一个神经元都是一个 TSFIS，输出层只有一个神经元，也是一个 TSFIS。当然也可以扩展为多输出，不同的输出间相互独立。

FIS: fuzzy inference system, 模糊推理系统，是一个完整的输入-输出模糊系统，比如上面介绍的 TS 模糊系统，就被称为 TSFIS

一个 TSFIS 神经元的模糊规则基（Fuzzy Rule Base, FRB）包含多条模糊规则，每条规则都包括前提部分和结论部分。一阶 TSFIS 的结论是输入的线形方程。FRB 的规则形式如下

$$R_i^h : \quad \text{IF } x_1 \text{ is } G_{1,i} \text{ AND } \cdots \text{ AND } x_D \text{ is } G_{D,i} \\ \text{THEN } y \text{ is } y_i = p_{i,0} + p_{i,1}x_1 + \cdots + p_{i,D}x_D$$

D 是输入个数， x_d 是第 d 个输入分量 ($d = 1, \cdots, D$)。 R 是规则总个数 $G_{d,i}$ 是前提中相应的输入模糊隶属度函数 ($i = 1, \cdots, R$)。前提中采用 "AND" 作为模糊连接符。

一个 TSFIS 的参数即为输入前提模糊隶属度函数的参数和结论系数，二者的组合可表示特定输入的模糊结构。可采用多种模糊隶属度函数。采用不同的模糊连接符可以定义不同的模糊规则基。

整个网络包括如下参数：

- 模糊规则的前提（premise）中的输入隶属度的参数；
- 每一层的每一个 TS 模糊神经元的结论部分的输入系数；

一个 TS 模糊神经元（TSFN）建模出了一种输入的复杂函数，输入的隶属度函数代表了模糊区域，建模出了输入数据的不确定性。模糊区域可以表示语义标签。TSDFN 中的 TSFN 提取输入数据中的复杂模式，相应的 FRB 参数以模糊规则的形式表示模式的内部结构。

a TSFN in TSDFN extracts a complex pattern in input data and corresponding FRB parameters represent the internal structure of the pattern in the form of fuzzy rules.

4.2.2. 网络参数辨识

采用标准的误差反向传播来针对特定数据进行网络参数辨识。下面考虑一个一般的隐层 TSFN (S_h)，假设输入向量为 $\mathbf{x} = [x_1, x_2, \cdots, x_d, \cdots, x_D]$ 。

$\theta_{d,f}^h$ denotes parameter of f^th input MF of input d in premise part of a rule in FRB of S_h

θ^h 表示**某个**规则的输入隶属度函数的参数矩阵，那么

$$\boldsymbol{\theta}^h = \begin{bmatrix} \theta_{1,1}^h & \cdots & \theta_{1,f}^h & \cdots & \theta_{1,F}^h \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \theta_{d,1}^h & \cdots & \theta_{d,f}^h & \cdots & \theta_{d,F}^h \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \theta_{D,1}^h & \cdots & \theta_{D,f}^h & \cdots & \theta_{D,F}^h \end{bmatrix}$$

其中 F 是隶属度函数的参数个数（**个人解读**）。如果隶属度函数采用 **高斯** 函数，那么参数为均值和方差（参数的个数为 2）。为了进行反向传播，必须要计算梯度，因此隶属度函数必须是连续（可导？）的。

\boldsymbol{p}^h 表示结论部分的系数矩阵，那么

$$\boldsymbol{p}^h = \begin{bmatrix} p_{1,0}^h & \cdots & p_{1,f}^h & \cdots & p_{1,D}^h \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{r,0}^h & \cdots & p_{r,f}^h & \cdots & p_{r,D}^h \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ p_{R,0}^h & \cdots & p_{R,f}^h & \cdots & p_{R,D}^h \end{bmatrix}$$

其中 R 为规则个数。

对于输出层的 TSFN，其参数与隐层的 TSFN 类似，只不过将上标换为 O ，即 $\boldsymbol{\theta}^O, \boldsymbol{p}^O$ 。

给定输入，隶属度函数的输出表示为

$$\boldsymbol{\mu}^h = \begin{bmatrix} \mu_{1,1}^h & \cdots & \mu_{1,f}^h & \cdots & \mu_{1,D}^h \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mu_{r,1}^h & \cdots & \mu_{r,d}^h & \cdots & \mu_{r,D}^h \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \mu_{R,1}^h & \cdots & \mu_{R,d}^h & \cdots & \mu_{R,D}^h \end{bmatrix}$$

其中 $\mu_{r,d}^h = \mu_{G_{rd}^h}(x_d)$ 是第 h 个 TS 模糊神经元中第 r 个规则下第 d 个输入的隶属度。

规则的权重计算如下（t-norm?）

$$\begin{aligned} \omega_r^h &= \bigwedge_{d=1}^D \mu_{r,d}^h \\ \boldsymbol{\omega}^h &= [\omega_1^h \cdots \omega_r^h \cdots \omega_R^h]^T \end{aligned}$$

规则输出

$$y_r^h = p_{r,0}^h + p_{r,1}^h x_1 + \cdots + p_{r,d}^h x_d + \cdots + p_{r,D}^h x_D$$

$$\mathbf{y}^h = \mathbf{p}^h \times \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}$$

最终 S_h 的输出为

$$a^h = \frac{\sum_{r=1}^R \omega_r^h y_r^h}{\sum_{r=1}^R \omega_r^h}$$

该输出作为输出层的 STFN 的输入，经过上述类似的步骤可以得到一个输出，作为整个 TSFN-y 的输出。误差 $e = y - y_d$ 可用于 MSE 损失函数

$$J = \frac{1}{2n} \sum_{n=1}^N (e^n)^2$$

其中 N 是数据样本（输入输出对）总个数， e^n 是第 n 个样本对应的误差。

$$\frac{\partial}{\partial}$$

4.3. Trajectory Prediction

Multi-agent Trajectory Prediction with Fuzzy Query Attention. NIPS 2020.

5. 参考文献

无。