N93-29532-67
161816
p- 8

Adaptive Defuzzification for Fuzzy Systems Modeling

Ronald R. Yager and Dimitar P. Filev
Machine Intelligence Institute
Iona College
New Rochelle, NY 10801

ABSTRACT:

We propose a new parameterized method for the defuzzification process based on the simple M-SLIDE transformation. We develop a computationally efficient algorithm for learning the relevant parameter as well as providing a computationally simple scheme for doing the defuzzification step in the fuzzy logic controllers. The M-SLIDE method results in a particularly simple linear form of the algorithm for learning the parameter which can be used both off and on line.

1. Introduction

Recently with the intensive development of fuzzy control[1, 2], the problem of selection of a crisp representation of a fuzzy set, defuzzification has become one of the most important issues in fuzzy logic. In [3, 4] it was shown that the commonly used defuzzification methods,Center of Area (COA) and Mean of Maxima (MOM) [1, 2], are only special cases of a more general defuzzification method, called Generalized Defuzzification via **BA**sic **D**efuzzification **D**istribution (BADD). The BAD Distribution v_i, i=(1, n) of a fuzzy set D with membership function $D(x_i) = w_i$, $w_i \in [0, 1]$, is derived from its possibility distribution by use of the BADD transformation:

$$v_i = \frac{w_i^\alpha}{\sum_{j=1}^{n} w_j^\alpha} \quad , \alpha \geq 0 \qquad (1).$$

The BADD transformation converts the possibility distribution w_i to a probability distribution v_i, in a manner that preserves the features of D, $w_i > w_j \Rightarrow v_i \geq v_j$ and $w_i = w_j \Rightarrow v_i = v_j$. For $\alpha = 1$ the BADD transformation converts proportionally the possibility distribution w_i, i=(1, n) to BAD distribution v_i, i=(1, n). For $\alpha > 1$ it discounts the elements of X with lower grade of membership w_i. Through parameter α the BADD transformation relates the probability distribution v(x) to our confidence in the model [3, 4]. An increasing of α is associated with a decrease of uncertainty, decreasing of entropy and an increase in confidence. The defuzzified value obtained via the BADD approach is defined as the expected value of X over the BAD distribution v_i, i=(1, n):

$$d^{BADD} = \sum_{i=1}^{n} \frac{x_i w_i^\alpha}{\sum_{j=1}^{n} w_j^\alpha} , \quad \alpha \geq 0 \qquad (2)$$

It is evident, that for fixed α, the defuzzified value d^{BADD}, minimizes the mean square error, $E\{(x - d^{BADD})^2\}$. Thus the BADD defuzzified value is the optimal defuzzified value in the sense of minimizing the criterion

$$\sum_i (x_i - d^{BADD})^2 p_i \qquad (3).$$

The main conclusion of this approach was that the best defuzzified value in the sense of above criterion can be obtained by adaptation of parameter α *by learning.* Unfortunately the problem of learning the parameter α from a given data set using directly expression (2) is a constrained nonlinear programming problem and its solution is difficult in real control applications. In this paper we solve the learning problem by the introduction of a new transformation of the possibility distribution w_i, i=(1, n) to the probability distribution v_i, i=(1, n), called the Modified SemiLInear DEfuzzification (M-SLIDE) transformation. The introduction of this new transformation results in a simple linear expression for the defuzzified value involving one parameter. An algorithm for learning the parameter is proposed.

2. M-SLIDE Defuzzification Technique

Let the probability distribution u_i, i = (1, n) be obtained by the proportional transformation (normalization) of w_i,

$$u_i = c\, w_i = \frac{w_i}{\sum_{j=1}^{n} w_j}, \quad i = (1, n). \qquad (4)$$

The following transformation of the probability distribution u_i, i =(1, n) to a probability distribution v_i, i =(1, n) is defined as the M-SLIDE transformation:

$$v_i = \begin{cases} \frac{1}{m}[1 - (1 - \beta)\sum_{j \notin M} u_j] & \text{if } i \in M \\ (1-\beta)\, u_i & \text{if } i \notin M \end{cases} \qquad (5)$$

where m = card(M) is the cardinality of the set M of elements with maximal membership grades:

$$M = \{i \mid w_i = Max_j[w_j]\}$$

The derivation of the M-SLIDE transformation is expressed in detail in Yager & Filev [5]

The following theorem [5] shows some of the significant properties of the probability distribution obtained via the M-SLIDE transformation.

Theorem 1: Let w_i, i=(1,n) be the possibility distribution of a given fuzzy set and let v_i, i=(1,n) be obtained by application of transformations (4) followed by (5). Then it follows:

i. distribution v_i, i=(1,n) is a probability distribution;

ii. $w_i = w_j \Rightarrow v_i = v_j$, $\forall$ i,j=(1,n) (identity);

iii. $w_i > w_j \Rightarrow v_i \geq v_j$, $\forall$ i,j=(1,n) (monotonicity)

iv. $\beta = 0 \Rightarrow v_i = \frac{w_i}{\sum_{j=1}^{n} w_j}$, i=(1,n);

v. $\beta = 1 \Rightarrow v_i = 0$, $i \notin M$ and $v_i = \frac{1}{m}$, $i \in M$.

An immediate consequence of Theorem 1 is that the entropy of the M-SLIDE Distribution v_i, is maximal for $\beta = 0$ and minimal for $\beta = 1$.

When using the M-SLIDE transformation to obtain the probability distribution v_i the expected value,d, with respect to the elements x_i of support set is

$$d = \sum_{i=1}^{n} v_i x_i = (1-\beta) \sum_{i \notin M} u_i x_i + \frac{1}{m} [1 - (1-\beta) \sum_{i \notin M} u_i] \sum_{j \in M} x_j$$

$$d = (1-\beta) \sum_{i \notin M} u_i (x_i - d^{MOM}) + d^{MOM}$$

where d^{MOM} is the MOM defuzzified value,

$$d^{MOM} = \frac{1}{m} \sum_{j \in M} x_j.$$

It is evident that expected value d generalizes the MOM defuzzified value.

Definition 1. The process of selection of a deterministic value from the universe of discourse of a given fuzzy set by evaluation of the expected value d is called the Modified Semi LInear DEfuzzification (M-SLIDE) Method. The defuzzified value, denoted d^{MS}, obtained by application of the M-SLIDE method is called the M-SLIDE value and is defined as

$$d^{MS} = (1-\beta) \sum_{i \notin M} u_i (x_i - d^{MOM}) + d^{MOM}.$$

The next Theorem shows the relationship between the M-SLIDE method and the commonly used COA and MOM defuzzification methods.

Theorem 2. The M-SLIDE method reduces to the COA defuzzification method for $\beta = 0$ and to the MOM defuzzification method for $\beta = 1$.

Proof. For $\beta = 0$

$$d^{MS} = \sum_{i \notin M} u_i x_i + \frac{1}{m} m\, u_{max} \sum_{j \in M} x_j = \sum_{i \notin M} c\, w_i x_i + c\, w_{max} \sum_{j \in M} x_j$$

$$d^{MS} = \frac{1}{\sum_{j=1}^{n} w_j} [\sum_{i \notin M} w_i x_i + w_{max} \sum_{j \in M} x] = d^{COA}$$

where by d^{COA} we denote the defuzzified valued obtained by the COA defuzzification method.

For $\beta = 1$, $d^{MS} = d^{MOM}$.

Theorem 3. The following expressions of the M-SLIDE defuzzified value, d^{MS}, are equivalent:

$$d^{MS} = (1-\beta) \sum_{i \notin M} u_i (x_i - d^{MOM}) + d^{MOM}$$

$$d^{MS} = \beta \sum_{i \notin M} u_i (d^{MOM} - x_i) + d^{COA}$$

$$d^{MS} = \beta\, d^{MOM} + (1-\beta)\, d^{COA}$$

$$d^{MS} = \beta\, (d^{MOM} - d^{COA}) + d^{COA}$$

Proof. $$d^{MS} = (1-\beta) \sum_{i \notin M} u_i (x_i - d^{MOM}) + d^{MOM}$$

$$= \beta \sum_{i \notin M} u_i (d^{MOM} - x_i) + \sum_{i \notin M} u_i (x_i - d^{MOM}) + d^{MOM}$$

$$= \beta \sum_{i \notin M} u_i (d^{MOM} - x_i) + \sum_{i \notin M} u_i x_i - \sum_{i \notin M} u_i d^{MOM} + d^{MOM}$$

$$= \beta \sum_{i \notin M} u_i (d^{MOM} - x_i) + \sum_{i \notin M} u_i x_i - (1 - m\, u_{max})\, d^{MOM} + d^{MOM}$$

$$d^{MS} = \beta \sum_{i \notin M} u_i (d^{MOM} - x_i) + d^{COA}$$

$$= \beta \sum_{i \notin M} u_i d^{MOM} - \beta \sum_{i \notin M} u_i x_i + d^{COA}$$

$$= \beta (1 - m\, u_{max})\, d^{MOM} - \beta \sum_{i \notin M} u_i x_i + d^{COA}$$

$$= \beta d^{MOM} - \beta m\, u_{max} \frac{1}{m} \sum_{i \in M} x_i - \beta \sum_{i \notin M} u_i x_i + d^{COA}$$

$$= \beta d^{MOM} - \beta d^{COA} + d^{COA}$$

$$d^{MS} = \beta d^{MOM} + (1 - \beta)\, d^{COA} = \beta (d^{MOM} - d^{COA}) + d^{COA}$$

Theorem 3 provides convenient forms for the M-SLIDE defuzzified value as a linear function of the parameter β. In the next section we will use these forms for estimation of the parameter β in a learning procedure, capable of working on line.

3. Algorithm for Learning the M-SLIDE Parameter

In this section we solve the problem of learning the parameter β of the M-SLIDE method from a given sequence of fuzzy sets and desired defuzzified values. Furthermore we demonstrate that the M-SLIDE method can be used as an approximation of the Generalized Defuzzification Method via the BAD Distribution [3].

Assume we are given a collection of fuzzy sets U_k and the desired defuzzified values d_k, $k = (1, K)$. We denote by d_k^{MOM} and d_k^{COA} the defuzzified values of the fuzzy sets U_k under MOM and COA defuzzification methods. The problem of learning of the parameter β is equivalent to the recursive solution of the set of linear equations: $\beta * (d_k^{MOM} - d_k^{COA}) + d_k^{COA} = d_k$, $k = (1, K)$.

For simplification we denote: $c_k = d_k^{MOM} - d_k^{COA}$ and $y_k = d_k - d_k^{COA}$ and rewrite the set of equations that has to be solved in the form: $c_k \beta = y_k$ for $k = (1, K)$.

In general there is no guarantee that this set of equations can be exactly satisfied for some value of β and also that c_k doesn't vanish for some k. For this reason we seek a **least squares** solution of the set of equations under the assumption of noisy observation data. The solution of this classical mathematical problem can be obtained by the application of a number of different techniques . In this paper we shall use an algorithm that is a deterministic version of the well known Kalman filter [6] which is usually used to solve the same kind least squares of errors estimation problem for the case of dynamic systems.

The unknown parameter β that has to be estimated is regarded as a state vector of a hypothetical autonomous scalar dynamic system driven by the equations

$$\beta_{k+1} = \beta_k \text{ and } y_k = c_k \beta_k + \xi_k$$

where the term ξ_k denotes Gaussian white noise with covariance r_k. Then the recursive Kalman filter that gives the best estimate of the state vector β_k of this system has the form [6]:

$$\hat{\beta}_{k/k} = \hat{\beta}_{k/k-1} + g_k (y_k - c_k \hat{\beta}_{k/k-1}) \qquad \text{i}$$

$$\hat{\beta}_{k+1/k} = \hat{\beta}_{k/k} \qquad \text{ii}$$

$$p_{k/k-1} = p_{k-1/k-1} \qquad \text{iii}$$

$$g_k = p_{k/k-1} c_k \frac{1}{c_k^2 p_{k/k-1} + r_k} \qquad \text{iv}$$

$$p_{k/k} = p_{k/k-1} - g_k c_k p_{k/k-1} \qquad \text{v}$$

Roughly speaking the Kalman filter calculates at every step the best estimate of the state vector as a sum of the prediction of β at step k from its value at step k-1, $\hat{\beta}_{k/k-1}$, and a correction term proportional to the difference between current output value y_k and predicted output $c_k \hat{\beta}_{k/k-1}$. Equation iv calculates the varying gain, g_k, of the filter. The evolution of error covariance is given by equation v. Because of the static nature of the autonomous system $\hat{\beta}_{k+1/k} = \hat{\beta}_{k/k} = \beta_k$ and $p_{k/k-1} = p_{k-1/k-1} = p_{k-1}$ this significantly simplifies the algorithm to

$$\beta_k = \beta_{k-1} + g_k (y_k - c_k \beta_{k-1}) \qquad \text{(vi)}$$

$$g_k = p_{k-1} c_k \frac{1}{c_k^2 p_{k-1} + r_k} \qquad \text{(vii)}$$

$$p_k = p_{k-1} - g_k c_k p_{k-1} \qquad \text{(viii)}$$

by combining vi and vii a more compact form of the algorithm is obtained

$$\beta_k = \beta_{k-1} + p_{k-1} c_k \frac{1}{c_k^2 p_{k-1} + r_k} (y_k - c_k \beta_{k-1}) \qquad \text{(ix)}$$

$$p_k = p_{k-1} - p_{k-1}^2 c_k^2 \frac{1}{c_k^2 p_{k-1} + r_k} \qquad \text{(x)}$$

Because usually we have no idea about the magnitude of the additive noise ξ_k we shall consider $r_k = 1$. Then equation (x) is further simplified and we receive the following final form of the Kalman filter algorithm for recursive least square solution of the original set of equations :

$$\beta_k = \beta_{k-1} + \frac{p_{k-1} c_k}{c_k^2 p_{k-1} + 1} (y_k - c_k \beta_{k-1}) \qquad \text{xi}$$

$$p_k = \frac{p_{k-1}}{c_k^2 p_{k-1} + 1} \qquad \text{xii}$$

Regarding the initial conditions, it can be argued [7] that a reasonable assumption is to consider $\beta_0 = 0$ and nonnegative p_0.

The algorithm gives an unconstrained solution for β. Because of the requirement of β belonging to the unit interval, we shall restrict the solution β_k by applying a threshold to give the value β_k^* where

$$\beta_k^* = \begin{cases} 1 \text{ if } \beta_{k-1} + \Delta_k > 1 \\ 0 \text{ if } \beta_{k-1} + \Delta_k < 0 \\ \beta_{k-1} + \Delta_k \text{ otherwise} \end{cases}$$

where Δ_k denotes the second term in the right part of xi,

$$\Delta_k = \frac{p_{k-1}\, c_k}{c_k^2\, p_{k-1} + 1} (y_k - c_k \hat{\beta}_{k-1}).$$

The thresholding effect can be replaced by the following nonlinear expression:

$$\beta_k^* = 1 - 0.5\,[1 - 0.5\,(\beta_{k-1} + \Delta_k + |\beta_{k-1} + \Delta_k|) + |1 - 0.5\,(\beta_{k-1} + \Delta_k + |\beta_{k-1} + \Delta_k|)\,|\,]$$

The algorithm for learning the M-SLIDE parameter, based on Kalman filter, can now be summarized in the following.

<u>Algorithm for learning the parameter</u> β **(M-SLIDE Learning Algorithm)**

1. Set $\beta_0 = 0$; $p_0 > 0$.

2. Read a sample pair U_k, d_k.

3. Calculate: **i.** d_k^{MOM}; **ii.** d_k^{COA}; **iii.** $c_k = d_k^{MOM} - d_k^{COA}$; **iv.** $y_k = d_k - d_k^{COA}$

4. Update β_k, p_k: $\beta_k = \beta_{k-1} + \dfrac{p_{k-1}\, c_k}{c_k^2\, p_{k-1} + 1} (y_k - c_k\, \beta_{k-1})$ and $p_k = \dfrac{p_{k-1}}{c_k^2\, p_{k-1} + 1}$

5. Calculate β_k^* :

$$\beta_k^* = 1 - 0.5\,[1 - 0.5\,(\beta_{k-1} + \Delta_k + |\beta_{k-1} + \Delta_k|) + |1 - 0.5\,(\beta_{k-1} + \Delta_k + |\beta_{k-1} + \Delta_k|)\,|\,]$$

6. Update the current estimate of the parameter β: $\beta = \beta_k^*$.

We note that since the estimate of the parameter β is determined sequentially there is no need to resolve the whole set of equations when a new pair of data pair (U_{k+1}, d_{k+1}) becomes available for learning. The addition of a new data pair can be incorporated by just an additional iteration of the algorithm. This property of the algorithm allows it to be used for either off-line or on-line learning of the parameter β.

In the case when the *desired* defuzzified values, the d_k's, are the defuzzified values obtained from the defuzzification method using the BADD distribution, the Algorithm can be used to get an associated M-SLIDE parameter β corresponding to a BADD transformation parameter α.

The next example presents an application of the M-SLIDE learning algorithm.

Example. Assume our data consists of 10 fuzzy sets:

U_1 = {0/3, 0.6/4, 1/5, .8/6, 0.9/7, 0/8}; U_2 = {0/5, 0.9/7, 1/9, 1/11, 0.2/12, 0/13}; U_3 = {0/2, 0.4/3, 0.8/4, 1/5, 0.5/6, 0/7}; U_4 = {0/4, 1/5, 0.9/6, 1/7, 0.9/8, 0/9}; U_5 = {0/6, 0.3/7, 1/8, 0.6/9, 1/10, 0/11}; U_6 = {0/3, 0.2/4, 0.9/7, 1/9, 1/10, 0/12}; U_7 = {0/1, 0.9/4, 0.5/5, 1/7, 0.4/8, 0/10}; U_8 = {0/3, 0.5/7, 0.9/10, 1/11, 0.4/14, 0/16};

U_9 = {0/5, 0.2/6, 1/7, 1/9, 0.1/10, 0/11}; U_{10} = {0/4, 1/7, 0.5/8, 1/9, 0.7/10, 0/11}.

We used the BADD defuzzification method to generate the *ideal* defuzzified values, d_k, associated with each of these fuzzy sets. In this way we formed six different data sets, each consisting of 10 pairs (U_k, d_k) In each data set the d_k's where generated by a different BADD parameter α.

For each data set, using the M-SLIDE learning algorithm, we obtained the optimal estimate for the parameter β. The following tables show the results of the experimentation with our algorithm. In the tables below we note that d_k is the ideal value and d_k^c is the calculated defuzzification value using the M–SLIDE defuzzification procedure with the optimal estimated β parameter for that data set.

DATA SET # 1 OPTIMAL ESTIMATED β = 0.00022

k	1	2	3	4	5	6	7	8	9	10
d_k^c	5.60	9.26	4.59	6.47	8.79	8.42	5.82	10.39	7.91	8.43
d_k	5.60	9.26	4.59	6.47	8.79	8.42	5.82	10.39	7.91	8.43

DATA SET # 2 OPTIMAL ESTIMATED β = 0.10758

k	1	2	3	4	5	6	7	8	9	10
d_k^c	5.54	9.34	4.64	6.42	8.82	8.54	5.95	10.46	7.92	8.39
d_k	5.71	9.21	4.70	6.42	8.98	8.82	5.76	10.46	7.99	8.28

DATA SET # 3 OPTIMAL ESTIMATED β = 0.22539

k	1	2	3	4	5	6	7	8	9	10
d_k^c	5.47	9.43	4.68	6.37	8.84	8.66	6.09	10.53	7.93	8.34
d_k	5.72	9.32	4.77	6.37	9.00	8.93	5.88	10.58	8.00	8.15

DATA SET # 4 OPTIMAL ESTIMATED β = 0.66891

k	1	2	3	4	5	6	7	8	9	10
d_k^c	5.20	9.75	4.87	6.16	8.93	9.14	6.61	10.80	7.97	8.14
d_k	5.36	9.72	4.97	6.17	9.00	9.27	6.49	10.83	8.00	8.00

DATA SET # 5 OPTIMAL ESTIMATED β = 0.92394

k	1	2	3	4	5	6	7	8	9	10
d_k^c	5.05	9.94	4.97	6.04	8.98	9.42	6.91	10.95	7.99	8.03
d_k	5.08	9.94	5.00	6.04	9.00	9.45	6.88	10.96	8.00	8.00

DATA SET # 6 OPTIMAL ESTIMATED $\beta = 0.97293$

k	1	2	3	4	5	6	7	8	9	10
d_k^c	5.02	9.98	4.99	6.01	8.99	9.47	6.97	10.98	8.00	8.01
d_k	5.03	9.98	5.00	6.01	9.00	9.48	6.96	10.99	8.00	8.00

It is can be seen from the above example that the M-SLIDE learning algorithm learns values of the parameter β that allow a very good matching of the data set.

4. References

[1]. Lee, C. C., "Fuzzy logic in control systems: fuzzy logic controller, Part 1," IEEE Trans. on Systems, Man and Cybernetics 20, 404-418, 1990.

[2]. Lee, C. C., "Fuzzy logic in control systems: fuzzy logic controller, Part II," IEEE Trans. on Systems, Man, and Cybernetics 20, 419-435, 1990.

[3]. Filev, D. and Yager, R. R., "A generalized defuzzification method under BAD distributions," International Journal of Intelligent Systems 6, 687-697.

[4]. Yager, R. R. and Filev, D. P., "On the issue of defuzzification and selection based on a fuzzy set," Technical Report# MII-1201, Machine Intelligence Institute, Iona College, 1991.

[5]. Yager, R. R. and Filev, D. P., "SLIDE: A simple adaptive defuzzification method," IEEE Transactions on Fuzzy Systems, (To Appear).

[6]. Aoki, M., Optimization of Stochastic Systems, Academic Press: New York, 1967.

[7]. Deutsch, R., Estimation Theory, Prentice Hall: Englewood Cliffs, NJ, 1965.