Mass Assignment Theory for Personalisation, Bayes Nets and DataMining

J. F. Baldwin

Engineering Mathematics Dept.

University of Bristol

jim.baldwin@bristol.ac.uk

What is Personalisation

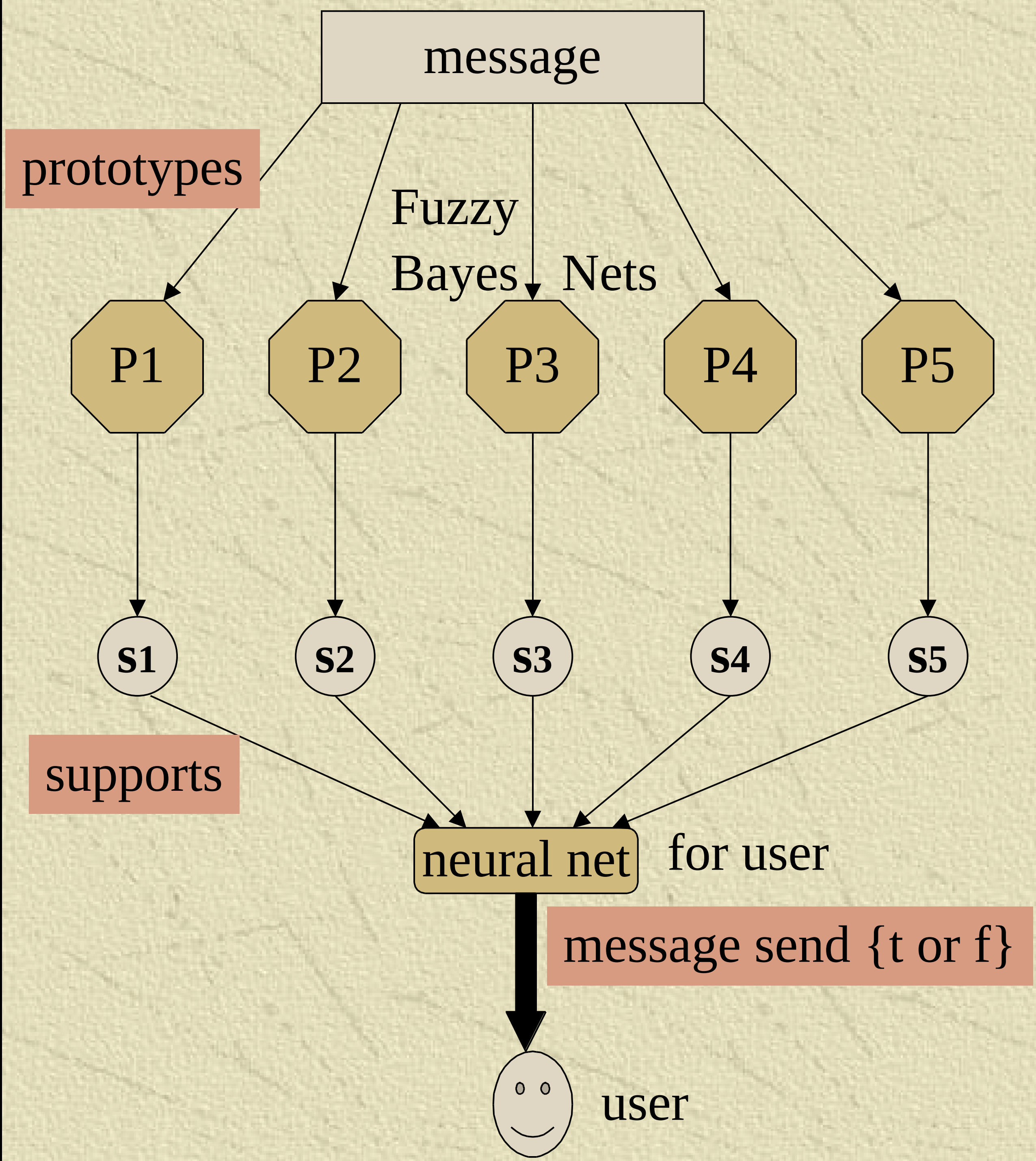
Personalisation - user profile and prototypes to provide:

What is received is what is required.

EXAMPLES
Web page retrieval to satisfy customer
Computer Interface
News Services, Advertising
Regulation of messages -SMS. E-Mail, voice
Personalised Data Mining

AI Machine Learning, Inference and Linguistics play a central role in providing the intelligent agents which can provide this personalisation service

Message Personalisation using Fuzzy Bayes Nets



Personalisation using Fril Evidential Logic Rules

Prototypes : use evidential logic rules
Neural Net expressed as evidential logic rules

Fril Evidential logic rule

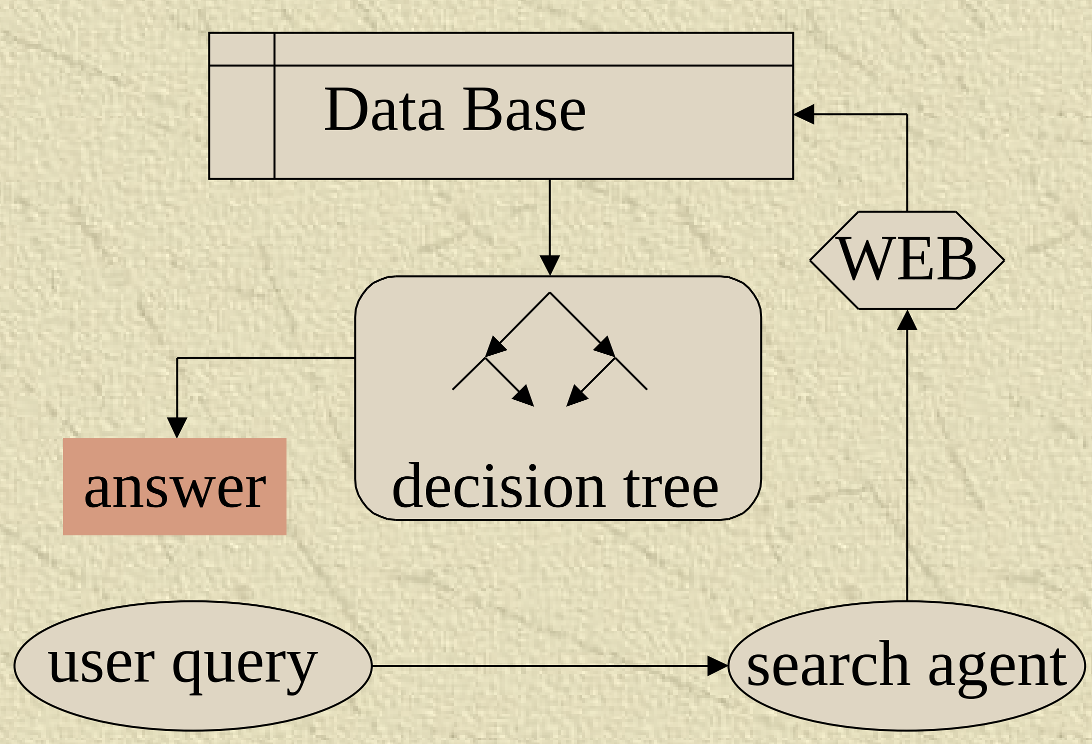
class is f IFF
x1 is g1 with weight w1
x2 is g2 with weight w2
. . . .
xn is gn with weight wn
through filter h

For input {xi = fi}
Let $\theta i = Pr(gi \mid fi)$

class is f with probability β
where
$\beta = \mu_h (w1\theta 1 + w2\theta 2 + \ldots + wn\theta n)$

f, gi, fi, h are fuzzy sets

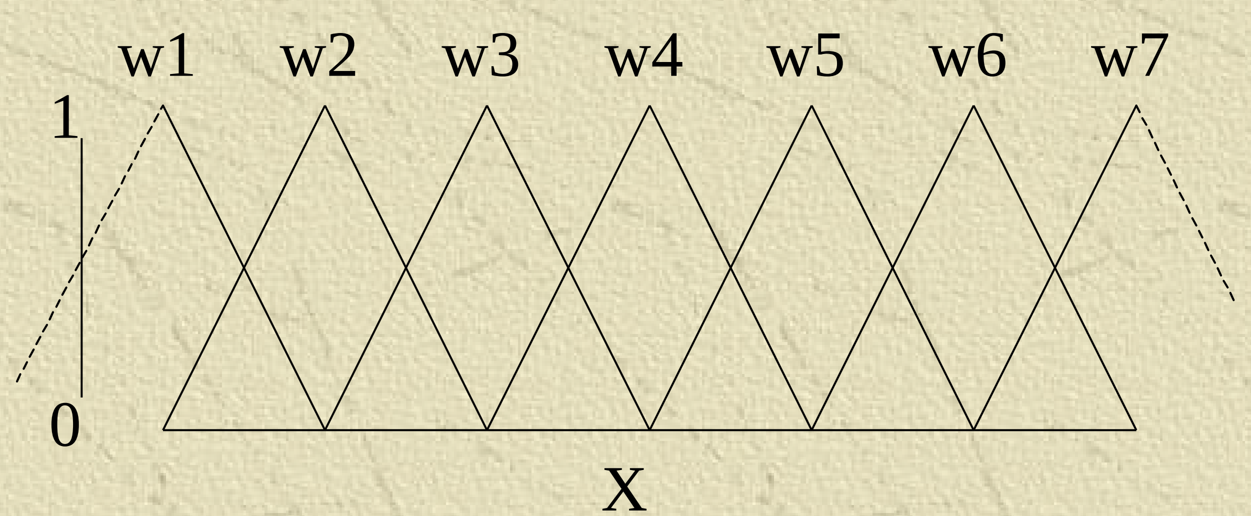
Data Mining Personalisation



Fuzzy Decision Tree can be used for classification and prediction.

Using fuzzy words provides good interpolation and compression and avoids over fitting.

Computing with Words

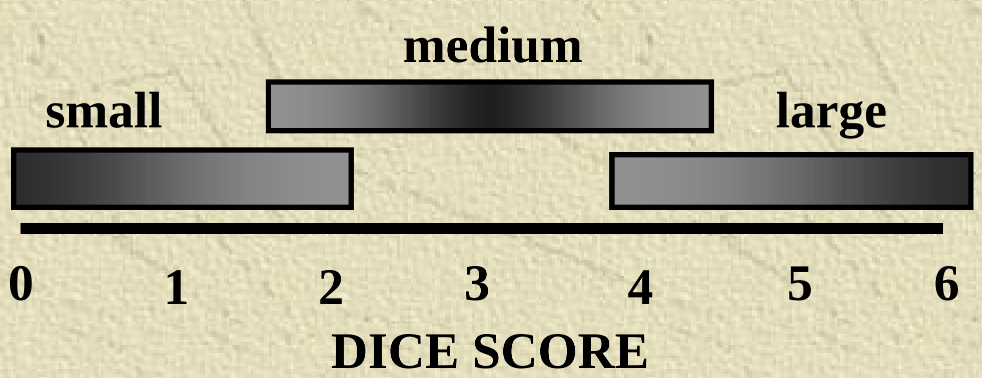


X replaced with {wi} - X can be discrete

x = wk / μ + w(k+1) / 1-μ for point value x
giving point value semantic unifications
Pr(wk | x) = μ, Pr(w(k+1) | x) = 1-μ
Pr(wi | x) = 0 for all i, i ≠ k, k+1
Generally x can be point, interval or fuzzy set.

Classical Algorithms modified to use these point semantic unification distributions.
Examples : ID3, Bayesian Nets, Neural Nets
Provides better interpolation and compression

Mass Assignment



Random Set of Voters % accept x as large	x	≤ 3	4	5	6
	%	0	20	80	100

large = 4 / 0.2 + 5 / 0.8 + 6 / 1

voters

1	2	3	4	5	6	7	8	9	10
6	6	6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5		
4	4								

constant threshold assumption : if voter accepts x and μy>μx then he must accept y

MA_{large} = {6} : 0.2, {5, 6} : 0.6, {4, 5, 6} : 0.2

Mass Assignment

Least Prejudiced Distribution

weighted dice is <u>small</u>
where
<u>small</u> = 1 / 1 + 2 / 0.7 + 3 / 0.3

prior for dice : 1:0.1, 2:0.1, 3:0.5, 4 :0.1, 5:0.1, 6:0.1

$MA_{\underline{small}}$ = {1} : 0.3, {1, 2} : 0.4, {1, 2, 3} : 0.3

Pr(dice is x | weighted dice is small) = probability that a randomly chosen voter chooses x as value of dice after being told dice value is <u>small</u>

Least Prejudiced Probability Distribution for dice value =
1 : 0.3 + 0.2 + 0.3(1/7) = 0.5429
2 : 0.2 + 0.3(1/7) = 0.2429
3 : 0.3(5/7) = 0.2143

For continuous fuzzy set we can derive a least prejudiced density function whose expected value can be used for defuzzification

Point Semantic Unification

weighted dice is small
where
small = 1 / 1 + 2 / 0.7 + 3 / 0.3

prior for dice : 1:0.1, 2:0.1, 3:0.5, 4 :0.1, 5:0.1, 6:0.1

Pr(x | small) = 1 : 0.5429, 2 : 0.2429, 3 : 0.2143

What is Pr(dice is about_2 | dice is small) ?
where
about 2 = 1 / 0.5 + 2 / 1 + 3 / 0.5

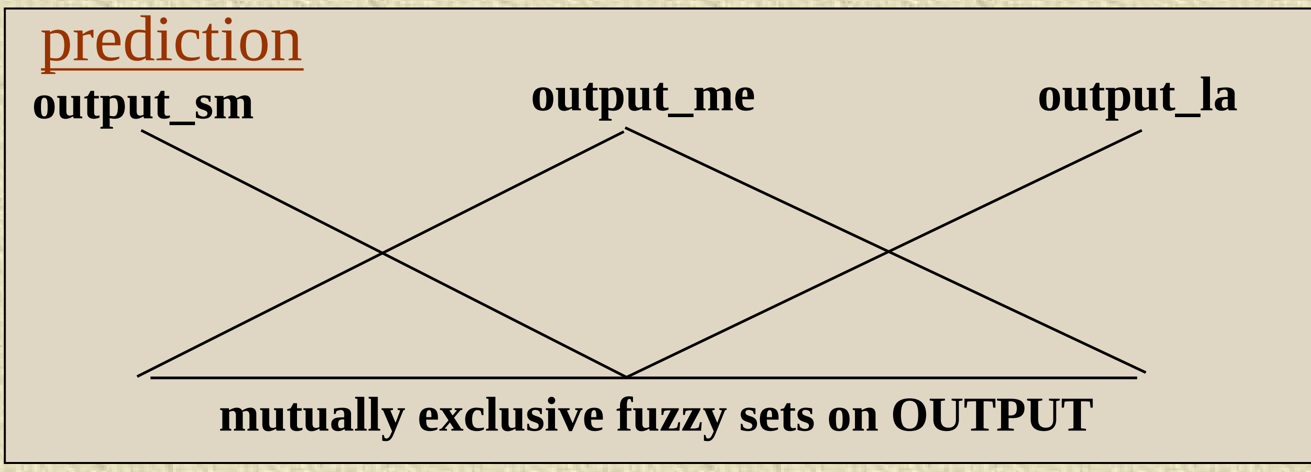
about_2 \ small	0.3 {1}	0.4 {1, 2}	0.3 {1, 2, 3}
0.5 {2}	0	1/2(0.2) = 0.1	1/7(0.15) = 0.0214
0.5 {1, 2, 3}	0.15	0.2	0.15

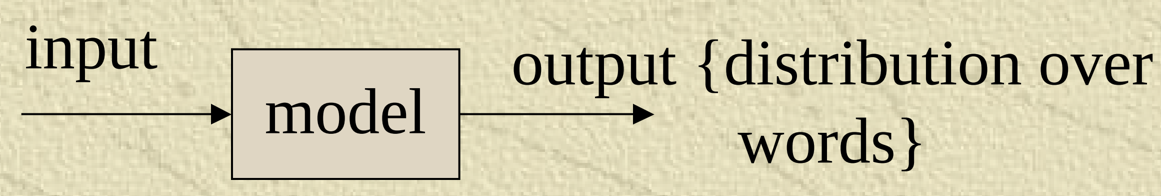
Pr(about_2 | small) = 0.6214

Prior
1 : 0.1
2 : 0.1
3 : 0.5
4 : 0.1
5 : 0.1
6 : 0.1

Point Semantic Unification used to determine Pr(f | g) where f is fuzzy set, g is point, interval or fuzzy set

Defuzzification





For one instance rules give supports

output_sm : θ1
output_me : θ2
output_la : θ3
1

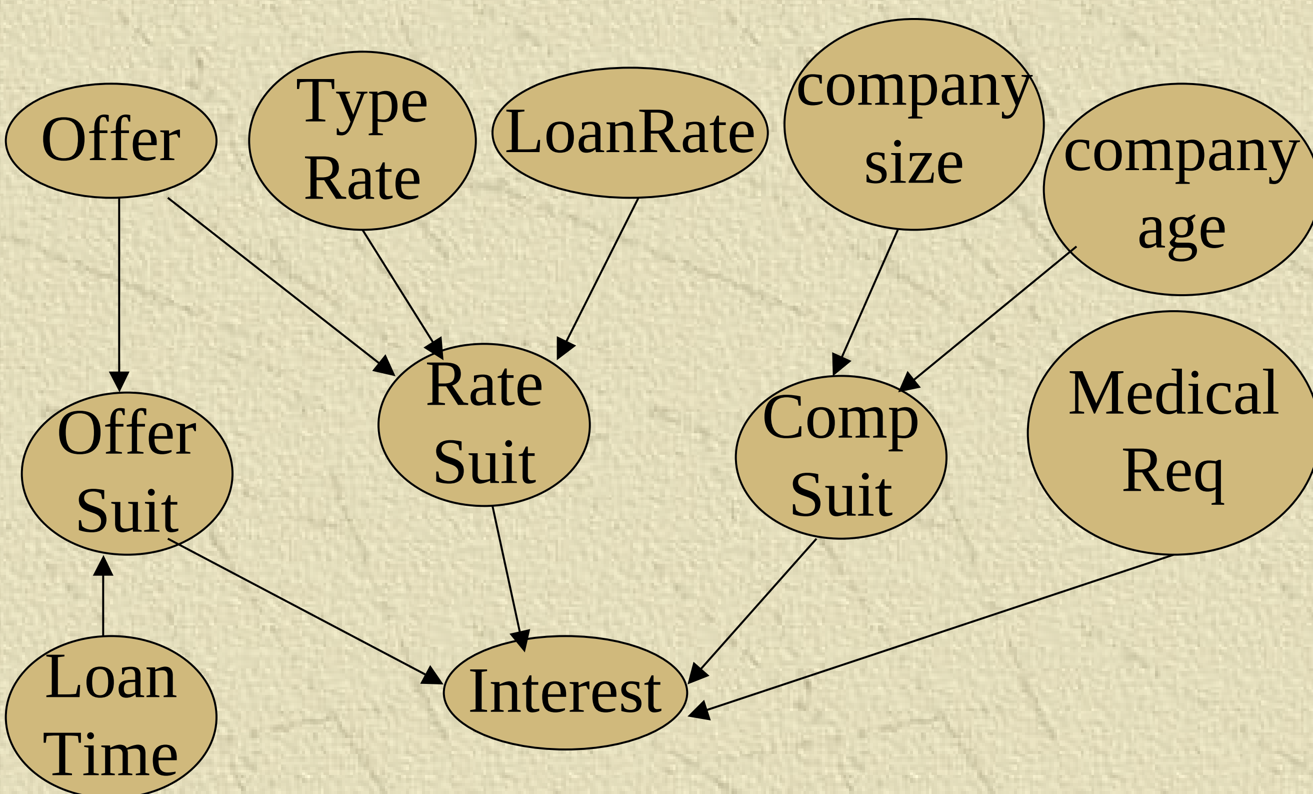
Distribution over words

Means of output_sm, output_me, output_la are μ1, μ2, μ3

Prediction = θ1μ1 + θ2μ2 + θ3μ3

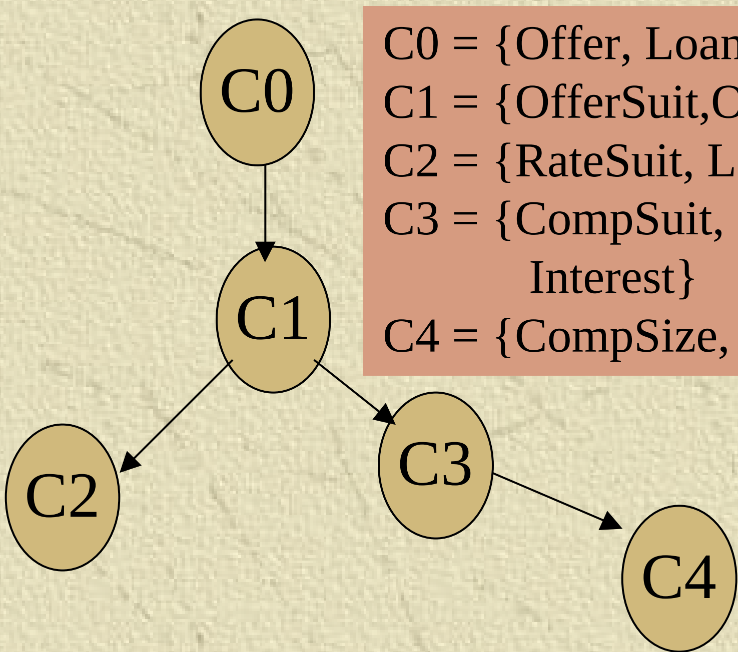
Defuzzification

A Fuzzy Bayesian Net



Offer : {mortgage, personal loan, car loan, credit card car insurance, holiday insurance, payment protection, charge card, home insurance}
Type Rate : {fixed, variable}
LoanRate : {good, fair, bad}
Company Size : {large, medium, small}
Company Age : {old, middle, young}
Loan Period : {long, medium, short}
Rate Suitable : {very, average, little}
Company Suitable : {very, average, little}
Medical req : {yes, no}
Offer Suit : {good, av, bad}
Interest : {high, medium, low}

Clique Tree and Message



C0 = {Offer, LoanPeriod, OfferSuit}
C1 = {OfferSuit,Offer, RateSuit}
C2 = {RateSuit, LoanRate, Offer, TypeRate}
C3 = {CompSuit, MedReq, OfferSuit, RateSuit, Interest}
C4 = {CompSize, AgeComp, CompSuit}

<u>Message :</u>
4% fixed rate long term mortgages available from 40 year old fairly large Company

FRIL

conceptual graph

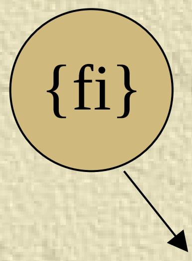
graph instantiations

Offer = mortgage, Type rate = fixed, loan time = long
CompSize = <u>dist</u>, CompAge = <u>dist.</u>

interest distribution → **defuzzified** → **s**

Translation

Bayes Node

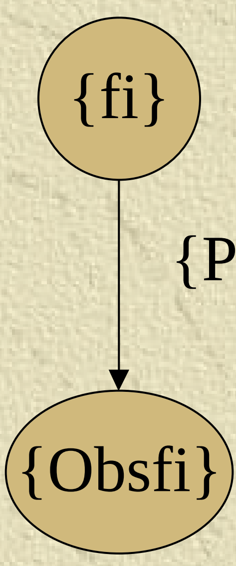


Variable X - value x
x is point or fuzzy set

Instantiate :
{fi} = {Pr(fi | x)}

Classical probability theory does not allow for this distribution update.

<u>Method 1</u>



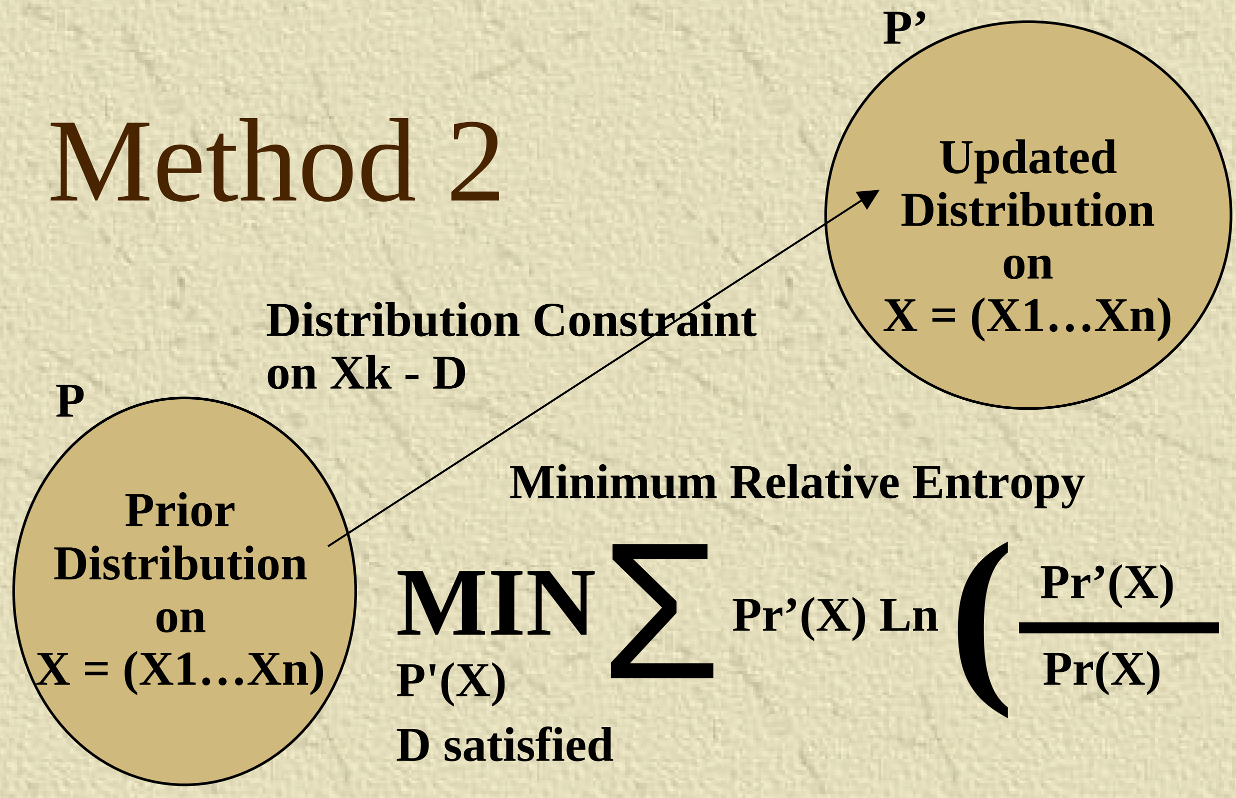
<u>likelihood node</u>

{Pr(approx_x | fi)} <u>method 2 better</u>

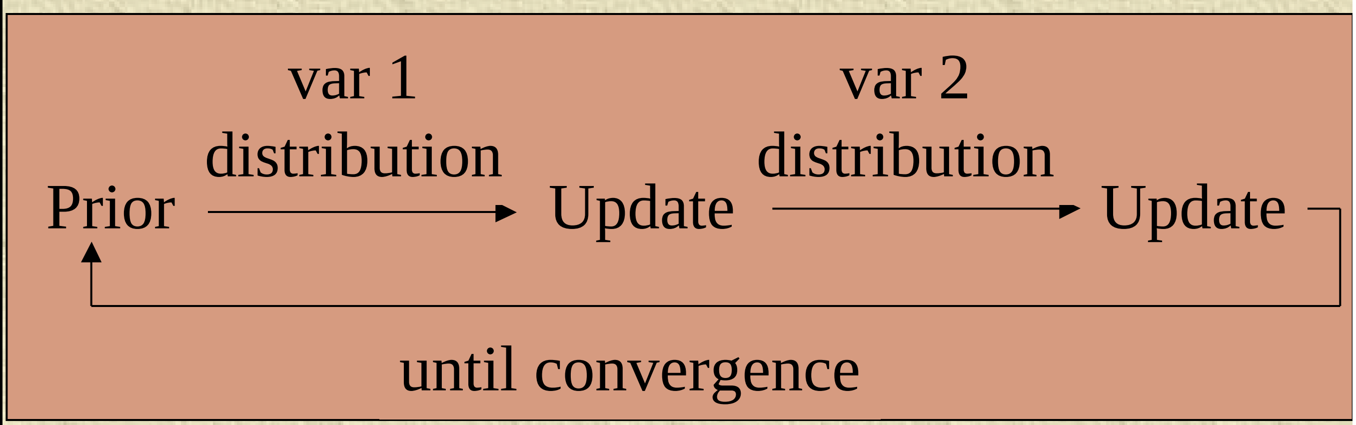
Observation node instantiated to approx_x

Update as normal

Method 2



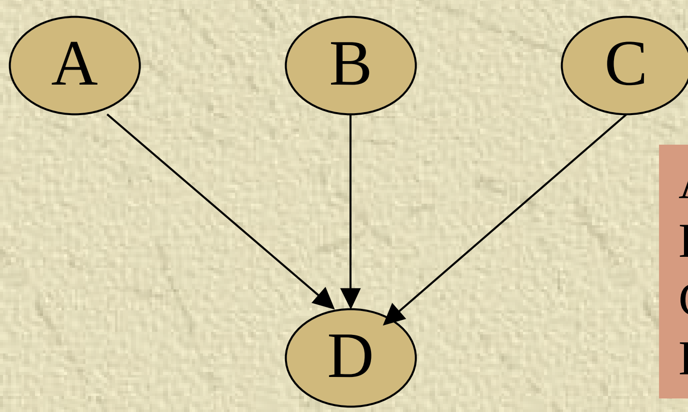
$$\underset{\substack{P'(X) \\ D\ \text{satisfied}}}{\text{MIN}} \sum Pr'(X)\ \text{Ln}\left(\frac{Pr'(X)}{Pr(X)}\right)$$



Message passing algorithms of Bayes Net
Compile to give prior probability distributions
Update this with variable instantiations

Bayes Net message passing algorithms for compile and updating modified to be equivalent to this modified updating

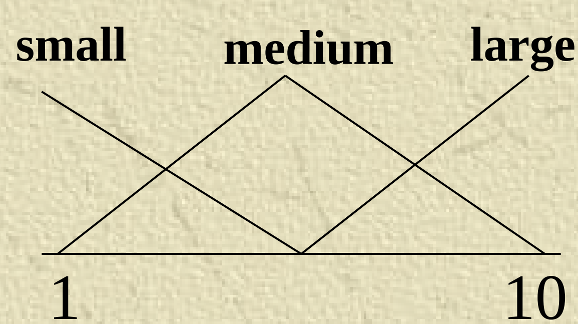
Learning Prototypes from Examples



A : {a1, a2, a3}
B : [1, 10]
C : {1, 2, 3, 4, 5, 6}
D : {d1, d2, d3}

	A	B	C	D
	a1	x	y	{d1, d2}

Fuzzify :
A : same as A above
B : {small, medium, large}
C : {low, av, high}
D : same as above



low = 1 / 1 + 2 / 0.8 + 3 / 0.4
av = 2 / 0.2 + 3 / 0.6 + 4 / 1 + 5 / 0.2
high = 5 / 0.8 + 6 / 1

Reduced Database

Semantic unification :

θ_1 = Pr(medium | x) θ_2 = Pr(large | x)
θ_3 = Pr(low | y) θ_4 = Pr(av | y)
Pr(d1) = 0.5   Pr(d2) = 0.5
Pr(small | x) = Pr(high | y) = 0
<u>Note</u>
x and y can be point values, intervals or fuzzy sets

<u>Reduced Data Base</u>

A	B	C	D	Joint Probability Distribution
a1	medium	low	d1	$.5\theta_1\theta_3$
a1	medium	low	d2	$.5\theta_1\theta_3$
a1	large	low	d1	$.5\theta_2\theta_3$
a1	large	low	d2	$.5\theta_2\theta_3$
a1	medium	av	d1	$.5\theta_1\theta_4$
a1	medium	av	d2	$.5\theta_1\theta_4$
a1	large	av	d1	$.5\theta_2\theta_4$
a1	large	av	d2	$.5\theta_2\theta_4$

Repeat for all lines of database

Calculate Pr(D | A, B, C)

Learning Architecture of Net for Data Mining

1. Search and Scoring based algorithms

2. Dependency Analysis algorithms

A. Node Ordering
B. Without node ordering

Complexity
(a) n^2
(b) n^4

Querying

Markov Cover :
Parents of query node + children of query node + parents of these children

Fril Rules - Prototype Model with Fuzzy fusion

((prototype p1 (offer Mortgage)) (evlog disjunctive **(**
(AgeOfCompany old_age**) 0.1**
(CompanySize large_size**) 0.15**
(LoanRate small_rate**) 0.4**
(TypeRate quite_fixed**) 0.1**
(LoanPeriod long_period**) 0.2**
(MedicalRequired bias_to_yes**) 0.05))) : ((1 1) (0 0))**

((prototype p1 (offer PersonalLoan)) (evlog disjunctive **(**
(AgeOfCompany old_age**) 0.1**
(CompanySize large_size**) 0.15**
(LoanRate medium_rate**) 0.4**
(TypeRate quite_variable**) 0.1**
(LoanPeriod short_period**) 0.2**
(MedicalRequired bias_to_no**) 0.05))) : ((1 1) (0 0))**

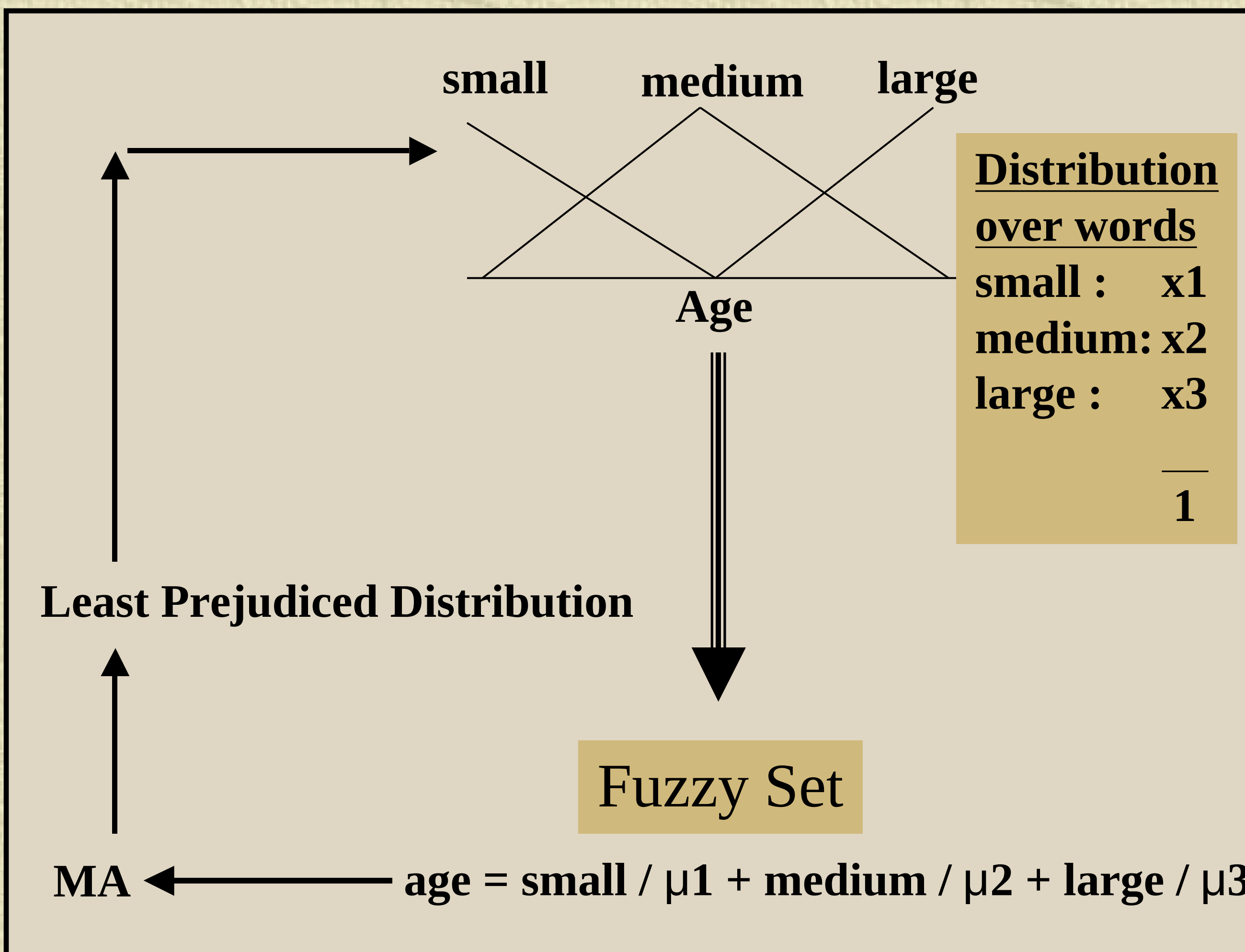
ETC - rules for other offers and other prototypes

((user u1 strong_accept**) (prot p1** high_supp**)**
(prot p2 high_supp**)) : ((1 1) (0 0))**
((user u1 weak_accept**) (prot p1** medium_supp**)**
(prot p2 medium_supp**)) : ((1 1) (0 0))**
((user u1 strong_reject**) (prot p1** low_supp**)**
(prot p2 low_supp**)) : ((1 1) (0 0))**

unbold - fuzzy sets

Learning Fuzzy Sets

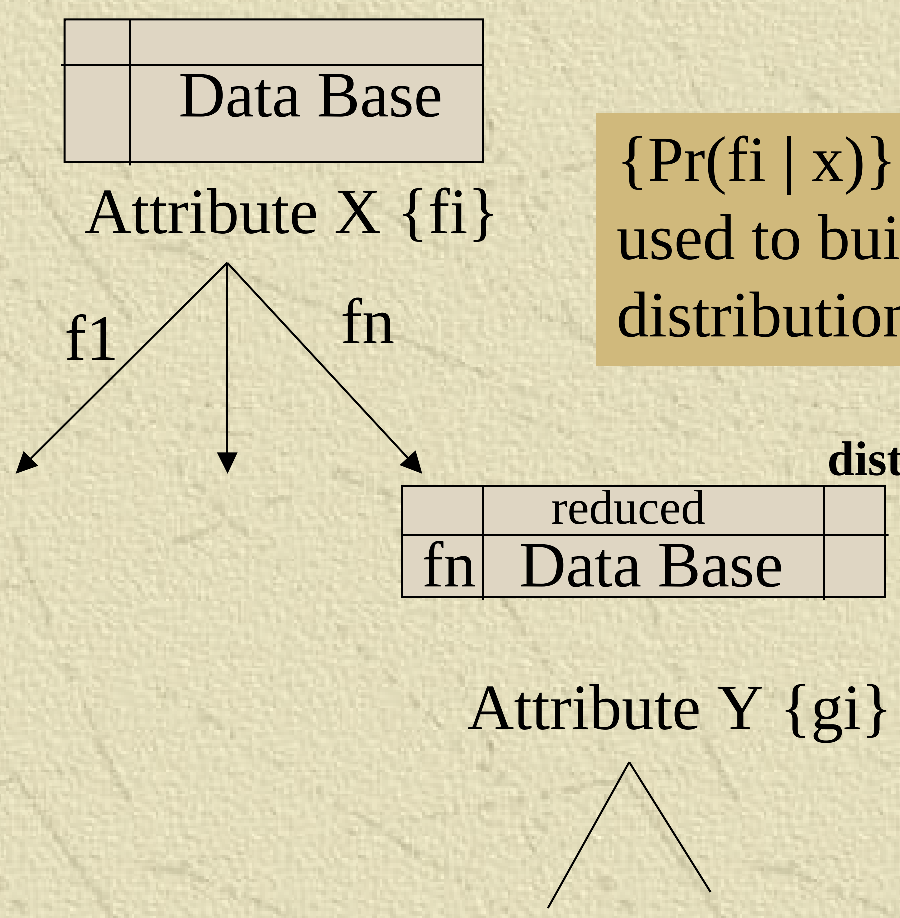
((prototype … **(evlog** disjunctive **(**
(AgeOfCompany age**) 0.1 . . .**



Weights in Evidential logic rules can also be learned.

Use of Evidential Logic Rules as given here emulates & extends maximal joins on fuzzy conceptual graphs

Fuzzy ID3



{Pr(fi | x)} used to build distributions

Entropy chooses order of attributes

Same algorithm as for classical ID3 except that distributions are recorded and used in future counting.

Final leaf nodes will give distributions over the required variable. Defuzzification used to give point value

ID3 for Learning to Fly - 1992

Human Pilots → **simulator** → **assigned flight plan**

(20 state variables, action) recorded
each time pilot took action
90, 000 examples

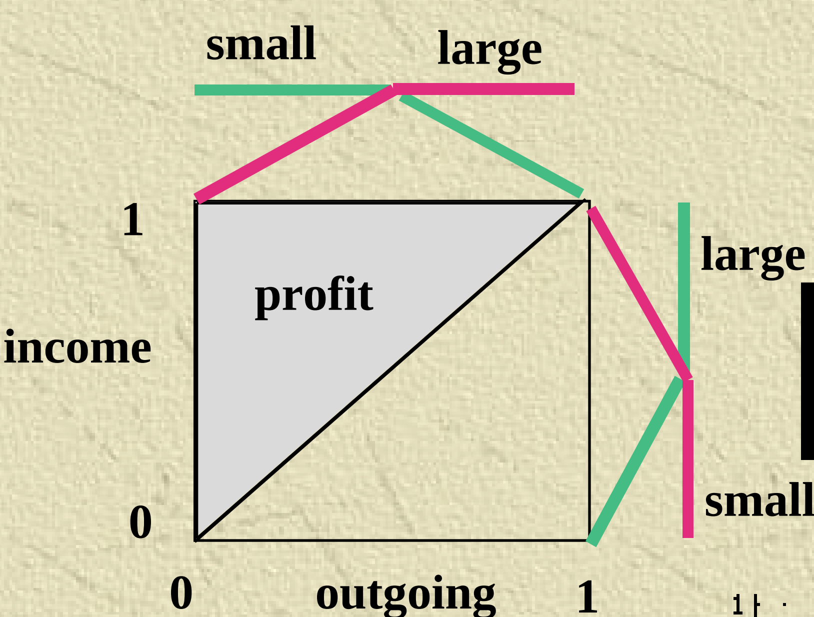
Data
↓
Knowledge

ID3 Decision Tree converted to rules
Rules hand coded as C program
Program put into control loop

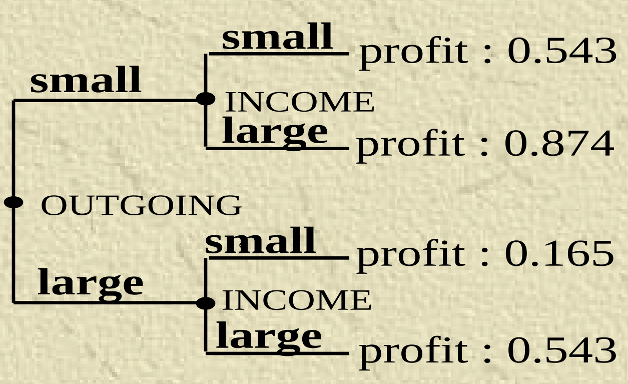
Program performed better than pilots

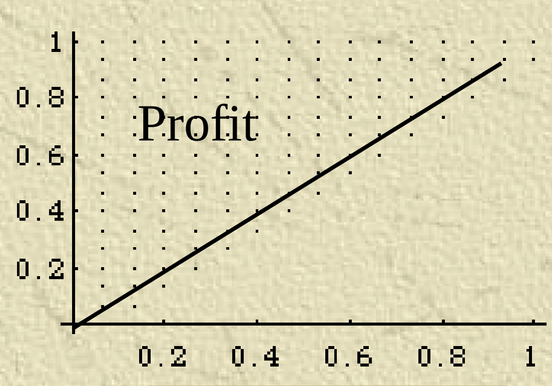
Use of Fuzzy ID3 would improve performance - better able to handle continuous variables and better able to smooth out noise

Fuzzy Sets important for Data Mining



Partition each universe with {small, large}





94.14% correct

Two crisp sets on each universe can give at most only 50% accuracy

We would require 16 crisp sets on each universe to give same accuracy as a two fuzzy set partition

SIN XY Prediction Example

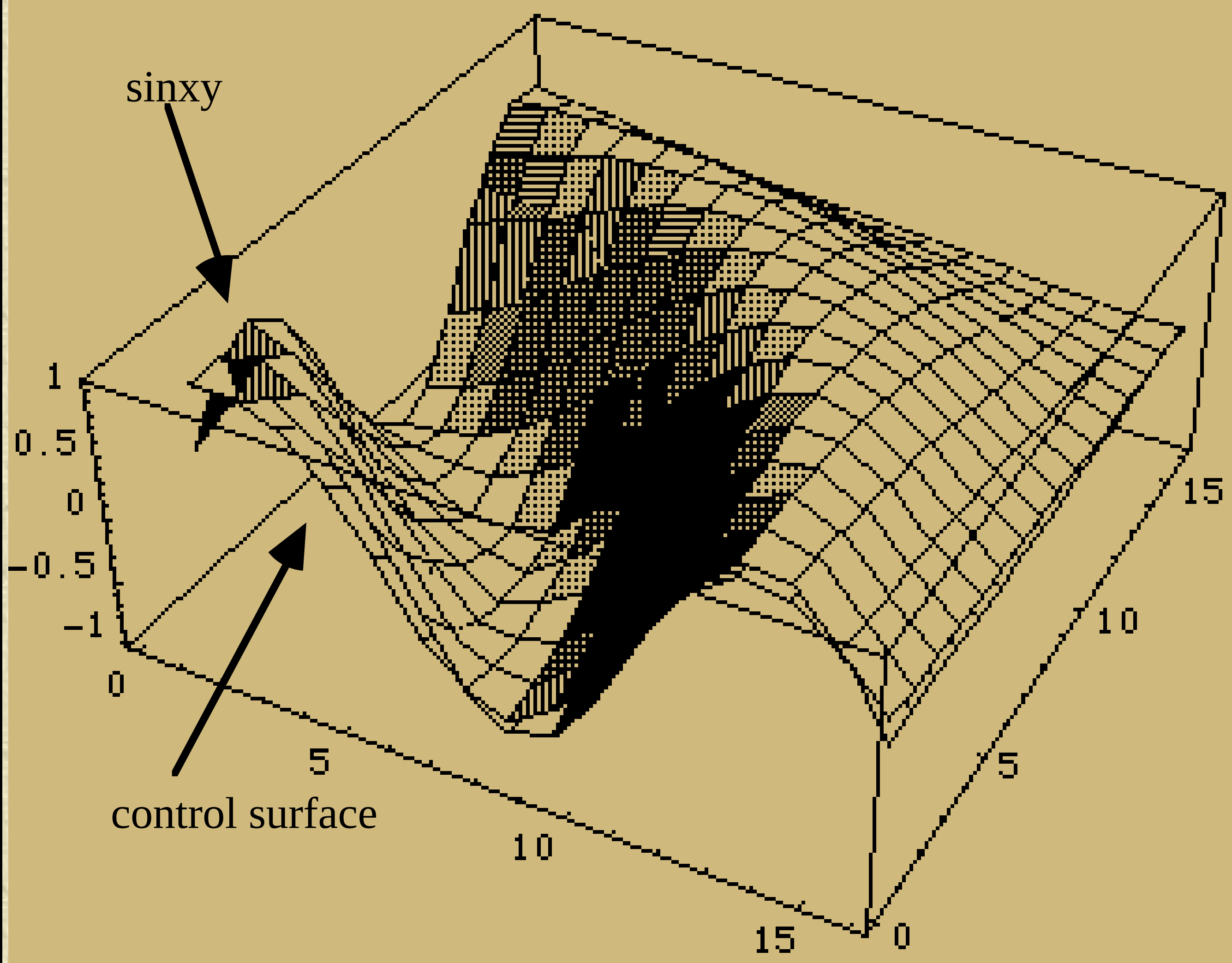
database
consists of 528 triples (X, Y, sin XY)
where the pairs (X, Y) form a regular grid on $[0, 3]^2$

about_ 0	= [0:1 0.333333:0]
about_0.3333	= [0:0 0.333333:1 0.666667:0]
about_ 0.6667	= [0.333333:0 0.666667:1 1:0]
about _ 1	= [0.666667:0 1:1 1.33333:0]
about_ 1.333	= [1:0 1.33333:1 1.66667:0]
about_1.667	= [1.33333:0 1.66667:1 2:0]
about _ 2	= [1.66667:0 2:1 2.33333:0]
about _2.333	= [2:0 2.33333:1 2.66667:0]
about _ 2.6667	= [2.33333:0 2.66667:1 3:0]
about _ 3	= [2.66667:0 3:1]

class_ 1	= [-1:1 0:0]
class _2	= [-1:0 0:1 0.380647:0]
class_ 3	= [0:0 0.380647:1 0.822602:0]
class_4	= [0.380647:0 0.822602:1 1:0]
class_5	= [0.822602:0 1:1]

Fuzzy ID3 decision tree with 100 branches

Percentage error of 4.22% on a regular test set of 1023 points.



Diabetes in Pima Indians

Diabetes mellitus in the Pima Indian population living near Phoenix Arizona.

Data

768 over 21 yrs females - 384 training, 384 test classes -
Attributes

1. Number of times pregnant
2. Plasma glucose concentration
3. Diastolic blood pressure
4. Triceps skin fold thickness
5. 2-Hour serum insulin
6. Body mass index
7. Diabetes pedigree function
8. Age

Each attribute space was partitioned by a uniform linguistic partition of 5 fuzzy sets with a 65% overlap.

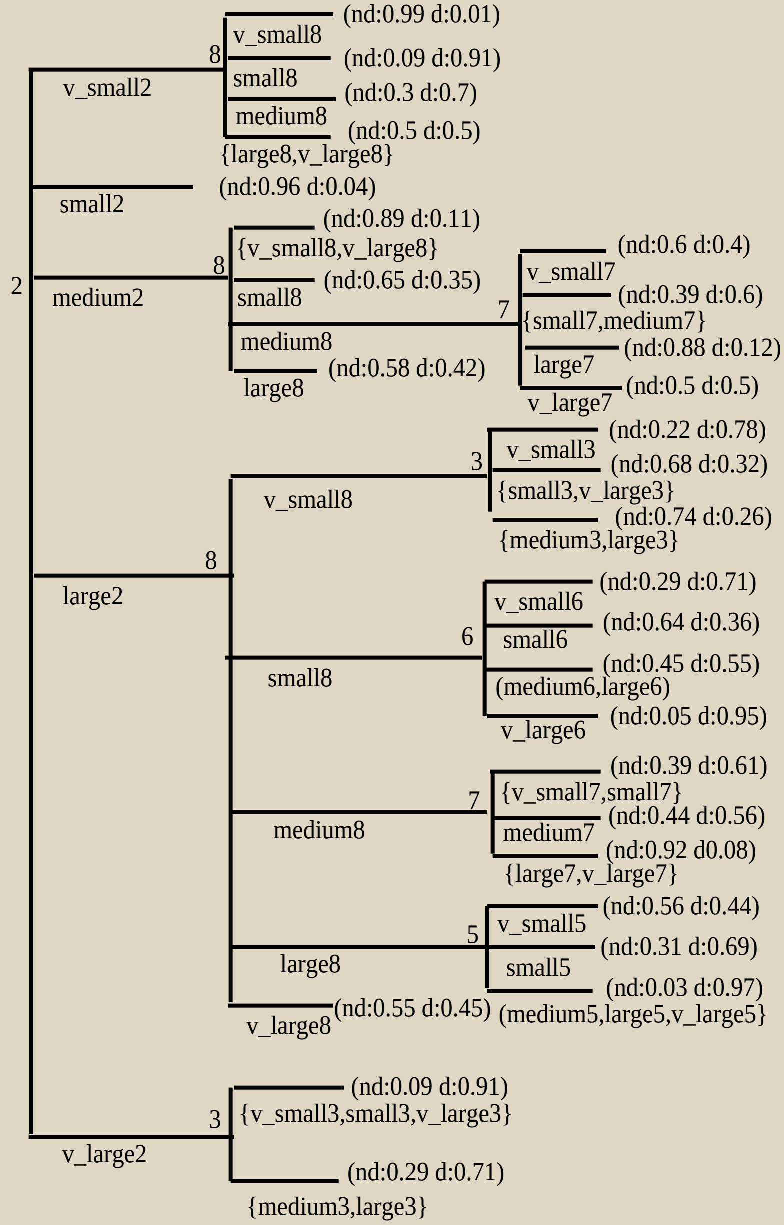
{very small, small, medium, large, very large} scaled for each attribute.

The decision tree was generated to a maximum depth of 4 given a tree of 161 branches. This gave an accuracy of 81.25% on the training set and 79.9% on the test set.

Forward pruning algorithm the tree complexity is halved to 80 branches. This reduced tree gives an accuracy of 80.46% on the training set and 78.38% on the test set.

Post pruning reduces the complexity to 28 branches giving 78.125% on the training set and 78.9% on the test set

Diabetes Tree



Decision Tree for Pima Indian Problem

A Control Example

Control given by fuzzy rules.
learned from reduced database using ID3

An Example Problem: The Van de Pol System

- The Van de Pol system is

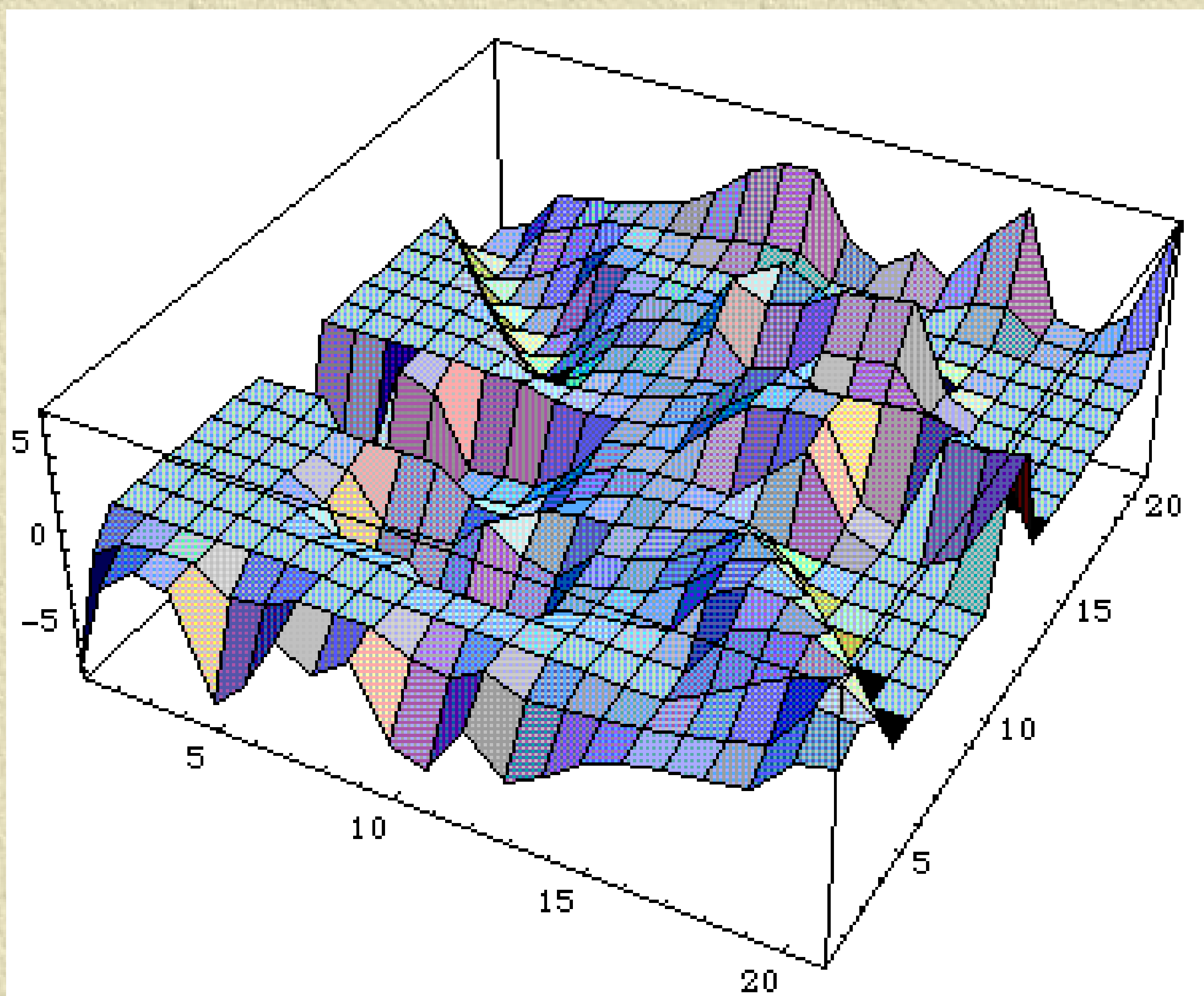
$$x_1' = x_2$$

$$x_2' = u + \varepsilon(1 - x_1^2)\, x_2 - x_1$$

- The control data was generated using an online control scheme introduce by J.F Baldwin in 1968.
- The data consists of a number of control paths in state space with starting points on a regular grid in $[-1\ 1]^2$
- The non-linearity parameter ε was set to 1.
- 20 fuzzy sets were used to partition both the state variable universes and the control universe and ID3
- was used to induce a rule base.

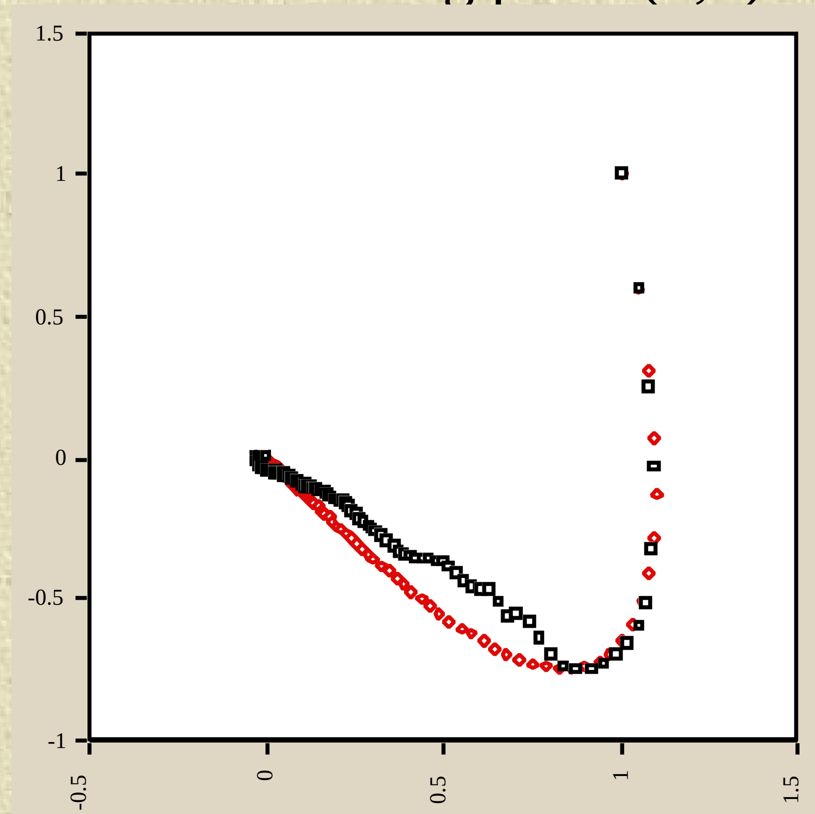
A Fuzzy Model

- The control surface for the ID3 derived model is.

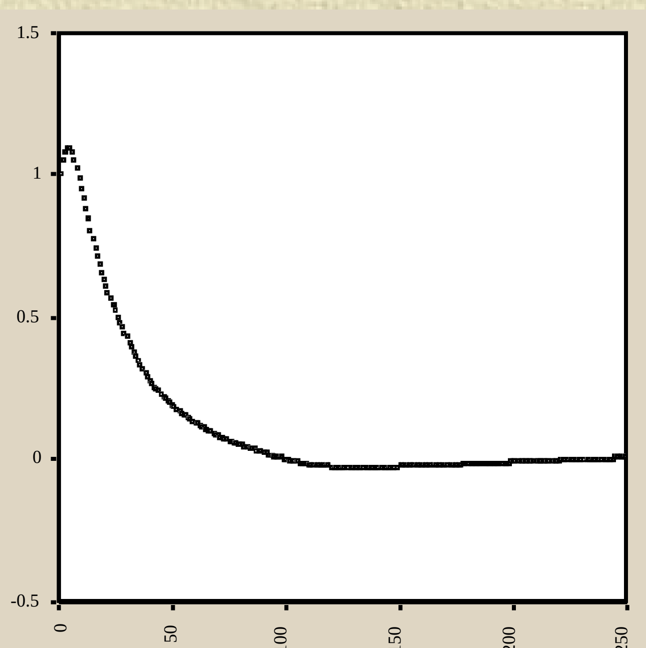


Comparing Control Paths

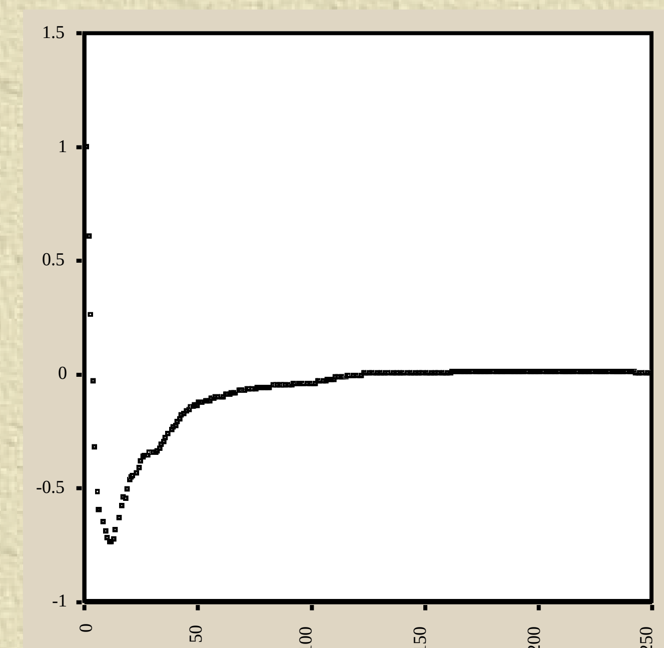
- Starting point (1,1)



rule method 1968 method



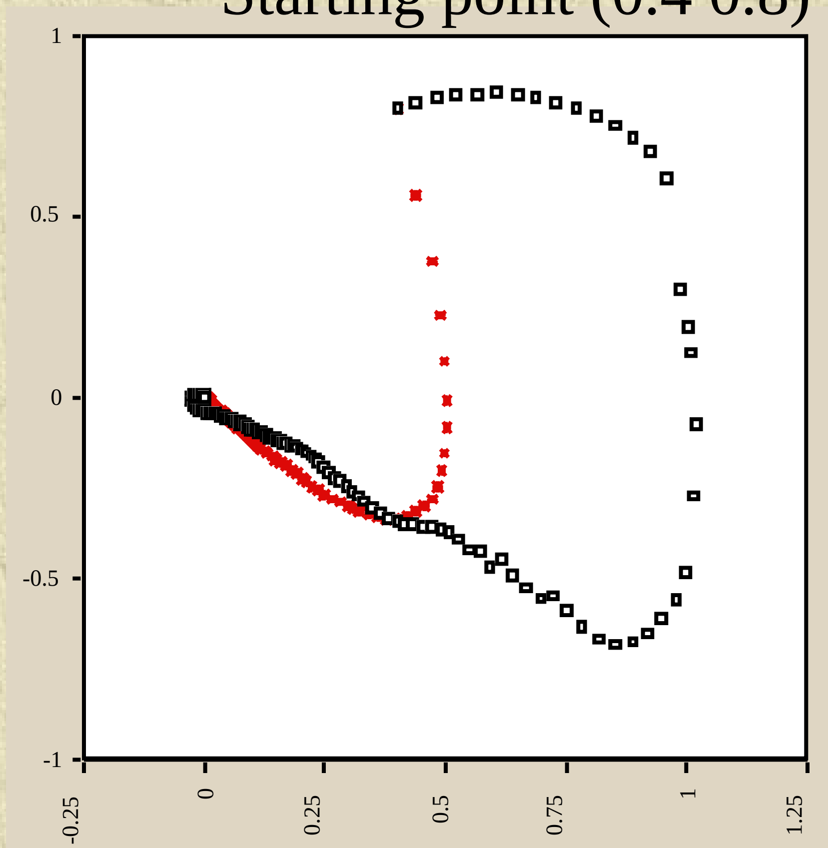
1968 method



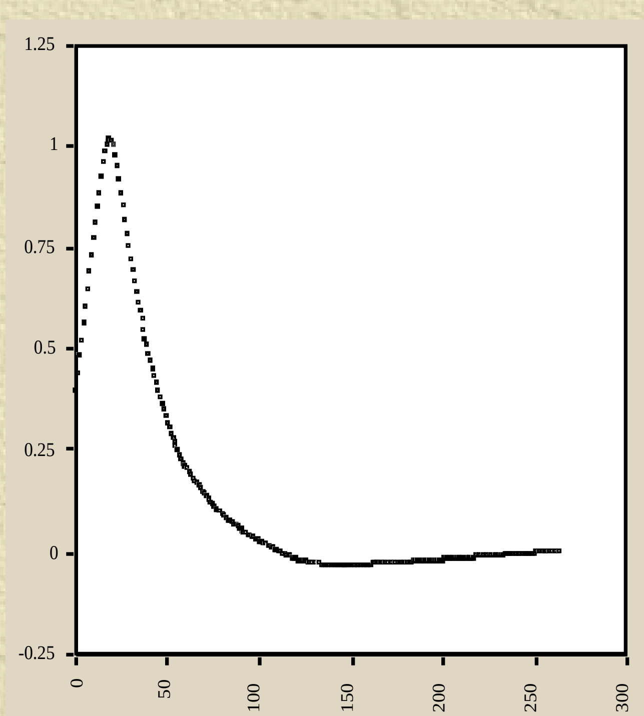
rule method

Another Comparison

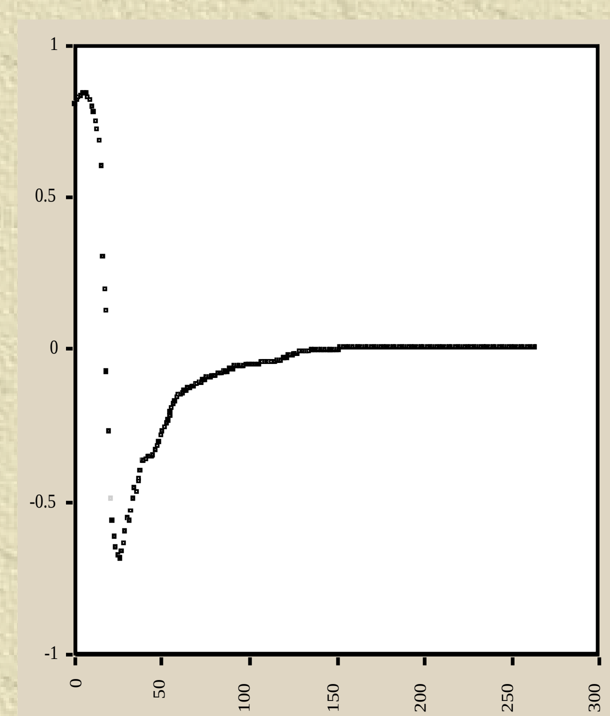
Starting point (0.4 0.8)



rule method   1968 method



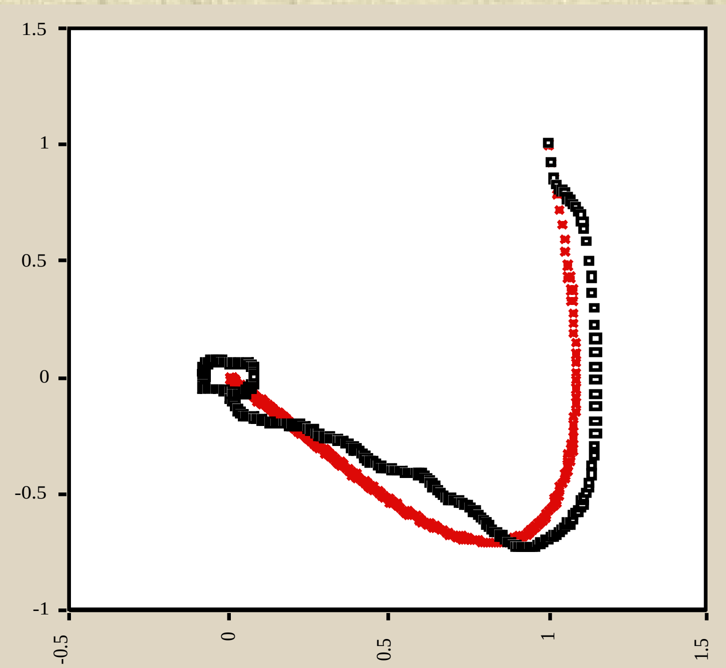
1968 method



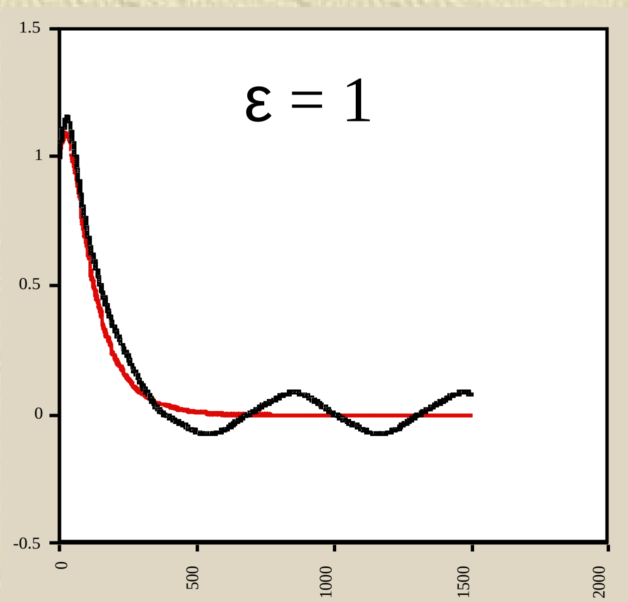
rule method

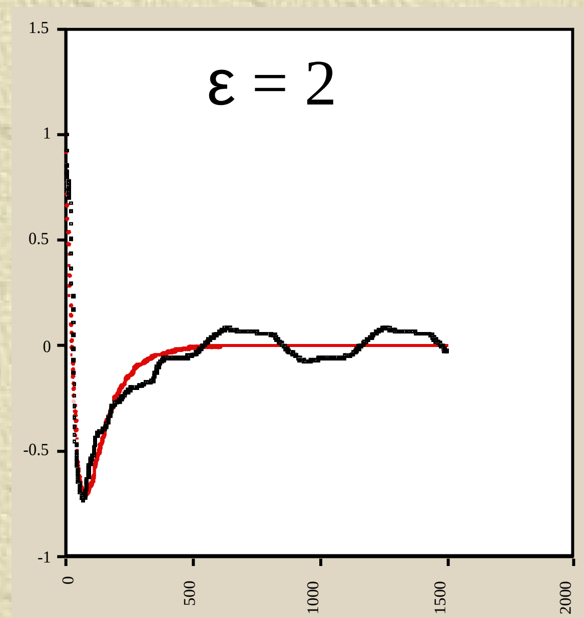
The Fuzzy Model is Robust

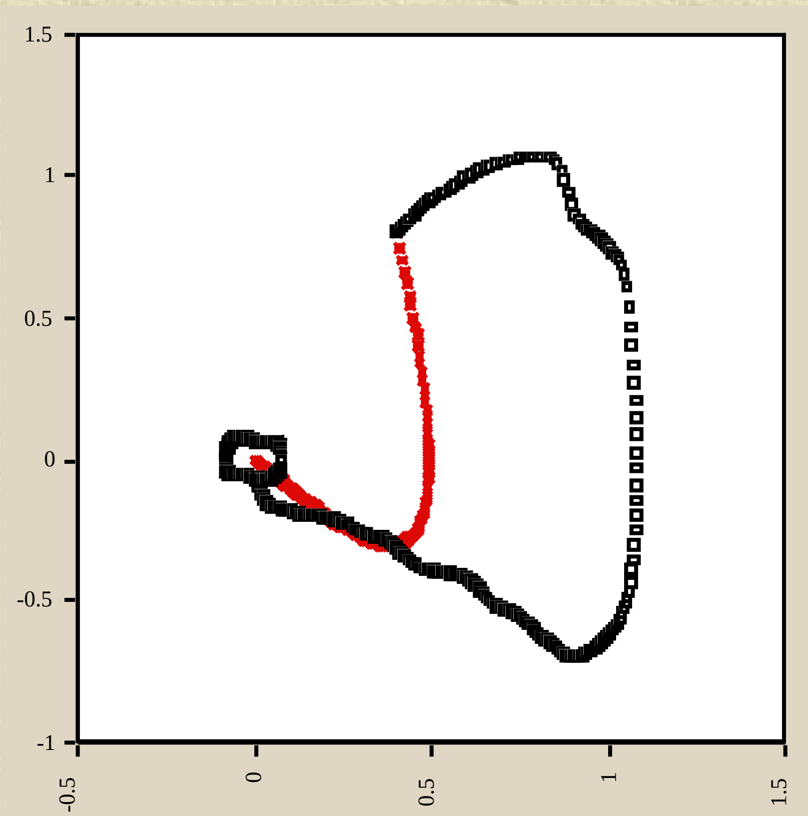
- Using the rule base learnt from a database where ε=1
- we attempted to control the system when ε=2.



- Starting point (1 1)







- Starting point
- (0.4 0.8)

rule method 1968 method