Removing the Black Box from Machine Learning

Instituto de Ciencias Físicas

2023

Angel Kuri-Morales
August 2023
akuri@itam.mx

Multivariate Approximation

Our purpose is to algebraically express the behavior of an dependent variable (y) as a function of a set of n independent variables ($v_1, v_2, \ldots, v_n$).

$$y = f(v_1, v_2, \ldots, v_n)$$

Data is assumed to be expressed in a table such as the one following:

A Multivariate Sample

V1	V2	V3	V4	V5	V6	F(V1,...,V6)
17.6924	-19.2393	13.7390	1.4218	-188.6625	2.7804	-172.2682
-0.2637	1.1110	0.0363	-0.0371	-0.0734	0.8846	1.6577
-4.3984	7.9895	9.1298	-0.8968	-29.5829	2.8539	-14.9049
-0.3881	-2.7033	-2.1130	0.2203	0.6413	2.7700	-1.5729
0.8088	8.4442	-1.0110	-0.1245	0.7551	2.5491	11.4217
-0.4151	6.2515	0.3410	-0.0602	-0.1874	2.1289	8.0587
15.1924	5.5425	-4.2747	-1.3692	156.0140	1.5804	172.6853
0.0320	-9.0747	0.0314	-0.0042	0.0010	2.4438	-6.5707
-0.7955	-0.2342	-0.2448	0.3648	2.1764	0.7327	1.9994
. . .	. . .	. . .	. . .	. . .	. . .	. . .
1.4102	4.2626	-1.0272	-0.2332	2.4664	1.8773	8.7561
-1.4069	-4.9818	-1.1969	-0.2326	-2.4539	2.0291	-8.2430
20.9671	-17.9575	15.4410	1.6984	-267.0812	2.6969	-244.2353
0.9949	8.8169	-0.6580	-0.1090	0.8133	2.1977	12.0557
1.4776	-3.8867	5.4001	0.5731	-6.3517	2.7455	-0.0420
0.2288	-0.5469	0.3504	0.1531	-0.2628	1.3530	1.2757
5.7689	-7.3154	12.0521	-1.2332	53.3540	2.7962	65.4226
-0.1644	10.8167	0.1573	-0.0196	-0.0241	2.5367	13.3027
-7.1679	9.7449	3.9029	-0.6777	-36.4344	2.1464	-28.4858
0.9709	2.1852	-2.4928	0.4210	-3.0651	2.1766	0.1957
-2.7095	2.4987	4.8167	0.9142	18.5768	2.0531	26.1499

The first issue we must tackle is: What form to adopt to achieve this goal?

The Stone-Weierstrass Approximation Theorem

"Every continuous function defined on a closed interval [a, b] can be uniformly approximated as closely as desired by a polynomial function".

Because polynomials are among the simplest functions, and because computers can directly evaluate polynomials, this theorem has both practical and theoretical relevance, especially in polynomial approximation. [1]

Experimental Models

Accordingly, the *approximant* is defined to have the form:

$$Y = c_1 X_1 + c_2 X_2 + \ldots + c_m X_m \qquad (1)$$

Where m is the number of desired *terms* and X_i is a *monomial* which denotes a product of the powers of the *n independent variables* each elevated to a maximum positive degree d, thus:

$$X_i = \Pi_{j=1}^{n} v_j^{k_j}; \quad 0 \le k_j \le d$$

Hyper-Parameters

To find our model the following hyper-parameters need to be defined:

a) The number of terms (i.e. monomials)

b) The powers of every monomial for every variable in $X_i = \Pi_{j=1}^{n} v_j^{k_j}; \quad 0 \le k_j \le d$

n is the number of variables

v_j if the *j-th* independent variable

j is the index of every independent variable

k_j is the degree of the *k-th* variable

d is the highest power of variables

Finding the Approximant once the Hyper-parameters have been defined

Knowing (a) and (b) above, the approximant

$$Y = c_1 X_1 + c_2 X_2 + \ldots + c_m X_m \qquad (1)$$

may be found using the so-called Ascent Algorithm (AA).

AA allows us to determine the coefficients of every monomial in (1) without imposing any particular set of conditions.

It minimizes the *minimax norm*, i.e. it finds the coefficients which yield the smallest absolute error between the known and approximated values

Example of an Algebraic Approximant

The following function

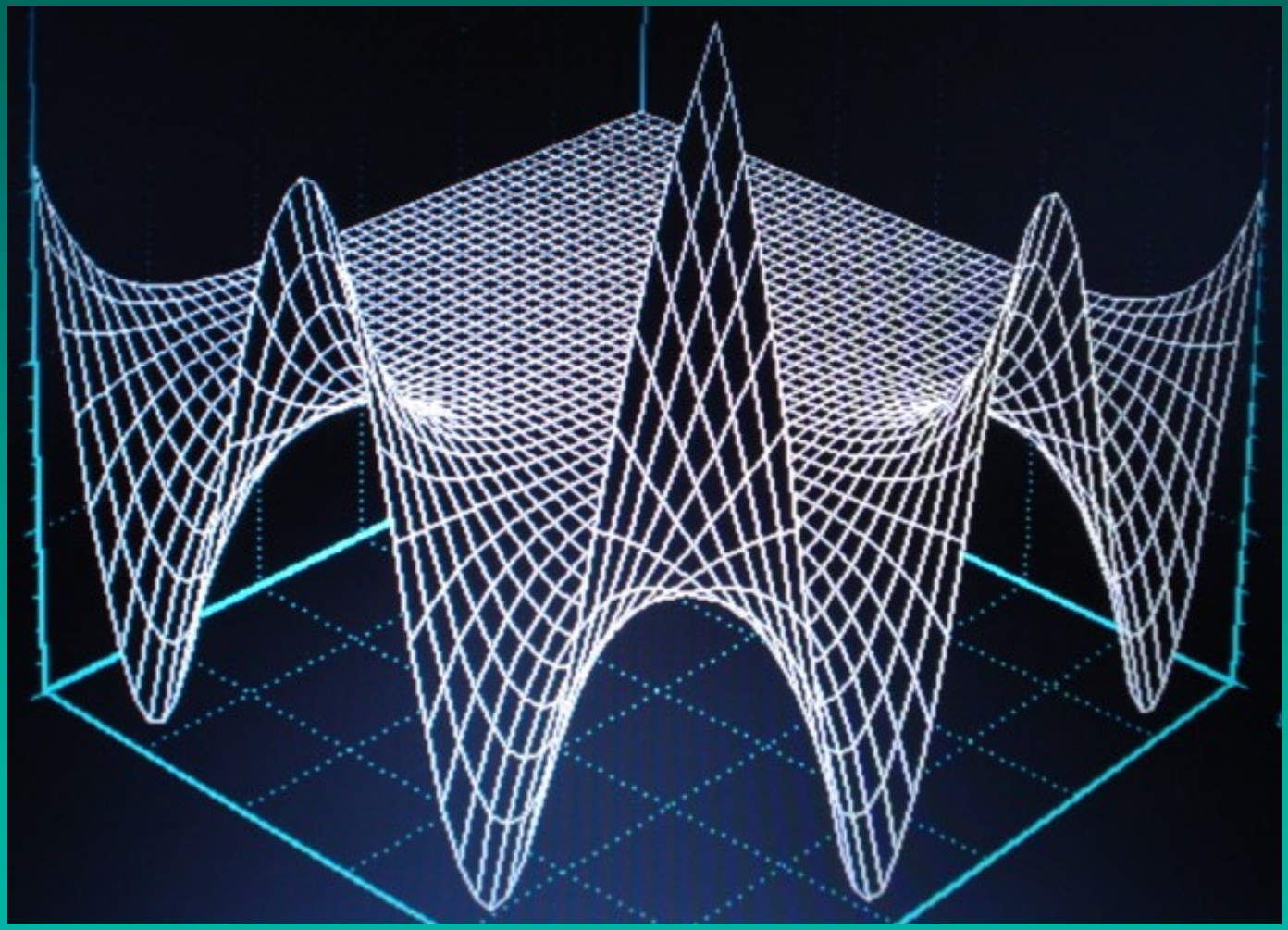
$$F(x,y)=e^{x}cos(y)+e^{y}cos(x)+ln(x)+y^{2}+1.12330837$$

was sampled and then it may be approximated with the following polynomial

$$F(x,y)\approx 0.5055-0.3576xy^{2}+3.2309x^{5}y-5.7738x^{8}y$$

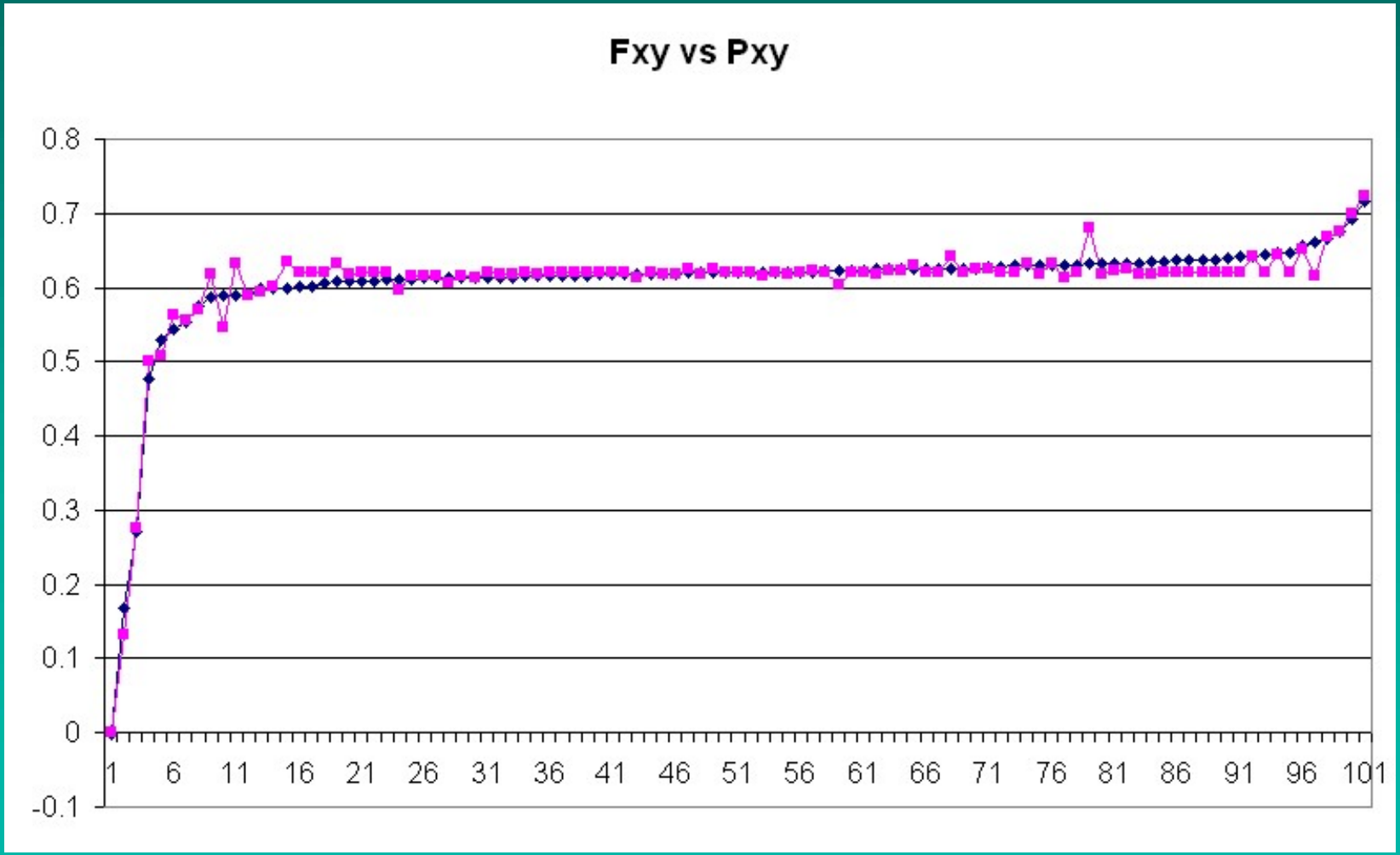
$$-18.8962x^{4}y^{5}+13.6171x^{3}y^{6}+7.4453x^{9}y^{2}$$

$f_{xy}(x,y)$ Partial View



The L_2 Bivariate Approximation

With an RMS error of 0.0996



The second issue we must tackle is: Which are the highest degrees of the approximant?

Cybenko´s Universal Approximation Theorem

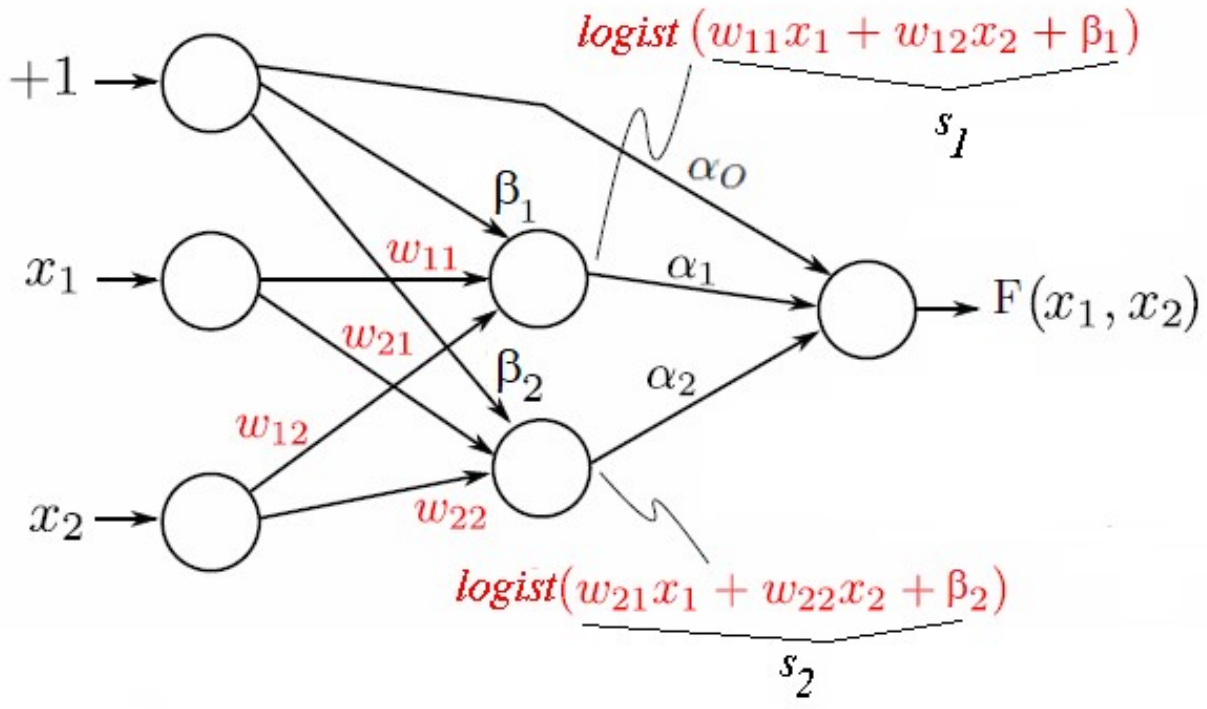
Theorem. Let $\varphi(\cdot)$ be a nonconstant, bounded and monotonically increasing continuous function. Let I_{m_O} denote the m_O-dimensional hypercube [0,1]. The space of continuous functions on I_{m_O} is denoted by $C(I_{m_O})$. Then, given any function $f \in C(I_{m_O})$ and $\varepsilon > 0$, there exist an integer M and sets of real constants α_i, β_i and w_{ij} where $i = 1, 2, \ldots, m_I$ and $j = 1, 2, \ldots, m_O$ such that we may define:

$$f(x_1, \ldots, x_{m_O}) = \alpha_0 + \sum_{i=1}^{m_I} \left[\alpha_i \cdot \varphi \left(\sum_{j=1}^{m_O} w_{ij} x_j + \beta_i \right) \right] \qquad (I.1)$$

as an approximate realization of the function f(·).

Illustration of Cybenko´s Theorem

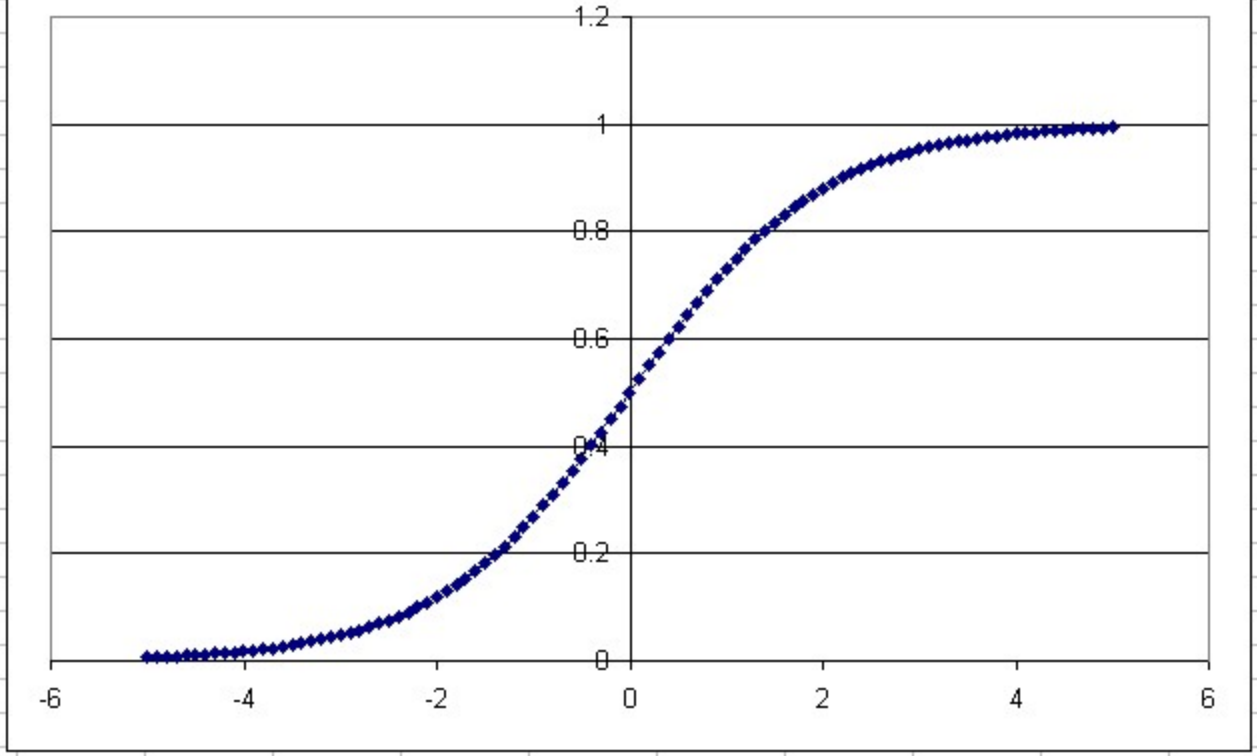
This theorem was designed to show that multi-layer perceptron networks were able to approximate any function



Approximating the Logistic Function

The logistic function may be approximated very closely by sampling its values from -5 to +5.

-5	0.006693
-4.9	0.007392
-4.8	0.008163
-4.7	0.009013
-4.6	0.009952
-4.5	0.010987
-4.4	0.012128
-4.3	0.013387
-4.2	0.014774
-4.1	0.016302
-4	0.017986
. .	. . .
4	0.982014
4.1	0.983698
4.2	0.985226
4.3	0.986613
4.4	0.987872
4.5	0.989013
4.6	0.990048
4.7	0.990987
4.8	0.991837
4.9	0.992608
5	0.993307



Approximating the Logistic Function

Notice that other than C_{00} all the even power monomials are approximately equal to zero. Then:

$$1/(1+e^{-x}) \approx C_{00} + \sum_{i=1}^{6} C_{2i-1} X^{2i-1} \qquad (I.3)$$

$C_{00} \approx +0.5000730019$

$C_{01} \approx +0.2490192529$

$C_{03} \approx -0.0194321579$

$C_{05} \approx +0.0014568548$

$C_{07} \approx -0.0000732019$

$C_{09} \approx +0.0000020441$

$C_{11} \approx -0.0000000235$

Universal Polynomial Approximation Theorem

We may then get an approximate polynomial version of Cybenko´s theorem

$$F(x_1,...,x_{mO}) = \alpha_0 + \sum_{i=1}^{m_I}\left[\alpha_i \cdot \varphi\left(\sum_{k=0}^{m_O} w_{ik}x_k\right)\right]$$

for convenience make $\nu_k = \sum_{k=0}^{m_O} w_{ik}x_k$,

and $F(x_1,...,x_{mO}) = \vartheta$

then:

$$\vartheta = \alpha_0 + \sum_{i=1}^{m_I}\left[\alpha_i \cdot logistic\left(\nu_k\right)\right]$$

$$\vartheta = \alpha_0 + \sum_{i=1}^{m_i}\left[\alpha_i \cdot (c_{00} + c_{01}\nu_k^1 .. + c_{11}\nu_k^{11})\right]$$

Powers of the Universal Polynomial Approximation

Powers of the Polynomial Expansion of the Hidden Layer	Powers of the Polynomial Expansion of the Output Layer	Powers of the Nested Polynomial's Expansion	Non Repeating Powers	Unique Powers
1	1	1	1	1
1	3	3	3	3
1	5	5		5
1	7	7	5	7
1	9	9		9
1	11	11	7	11
3	1	3		15
3	3	9	9	21
3	5	15		25
3	7	21		27
3	9	27	11	33
3	11	33		35
5	1	5	15	45
5	3	15		49
5	5	25	21	55
5	7	35		63
5	9	45	25	77
5	11	55	27	81
7	1	7		99
7	3	21	33	121
7	5	35		
7	7	49	35	
7	9	63		
7	11	77	45	
9	1	9		
9	3	27	49	
9	5	45	55	
9	7	63		
9	9	81	63	
9	11	99		
11	1	11	77	
11	3	33		
11	5	55	81	
11	7	77	99	
11	9	99		
11	11	121	121	

Universal Approximation Theorem

Table 1. Combinations of unique odd powers of the terms of the expansion of *logistic(x)*

List L				Indices of the Elements of L			
1	11	33	63	**1**	**6**	**11**	**16**
3	15	35	77	**2**	**7**	**12**	**17**
5	21	45	81	**3**	**8**	**13**	**18**
7	25	49	99	**4**	**9**	**14**	**19**
9	27	55	121	**5**	**10**	**15**	**20**

Universal Approximation Theorem

Therefore, finding the best multivariate model for (1) reduces to determine

a) The number of terms,

b) The powers of every variable v_j in term $\prod_{j=1}^{n} v_j^{k_j}$ provided that $\sum_{j=1}^{n} k_j \in L(i)$;

c) The associated coefficients so that the approximation error is minimized.

Optimization Goal

The goal of polynomial multivariate approximation may thus be stated as

"Find the coefficients of the linear combination of a set of *m* monomials $X_i = \Pi_{j=1}^{n} v_j^{k_j}; \sum_{j=1}^{n} k_j \in L(i);$ such that the minimax approximation error is minimized" (2)

The third issue we must tackle is: **How do we find the values of the powers and coefficients? of** $X_i = \Pi_{j=1}^{n} v_j^{k_j}; \sum_{j=1}^{n} k_j \in L(i)$

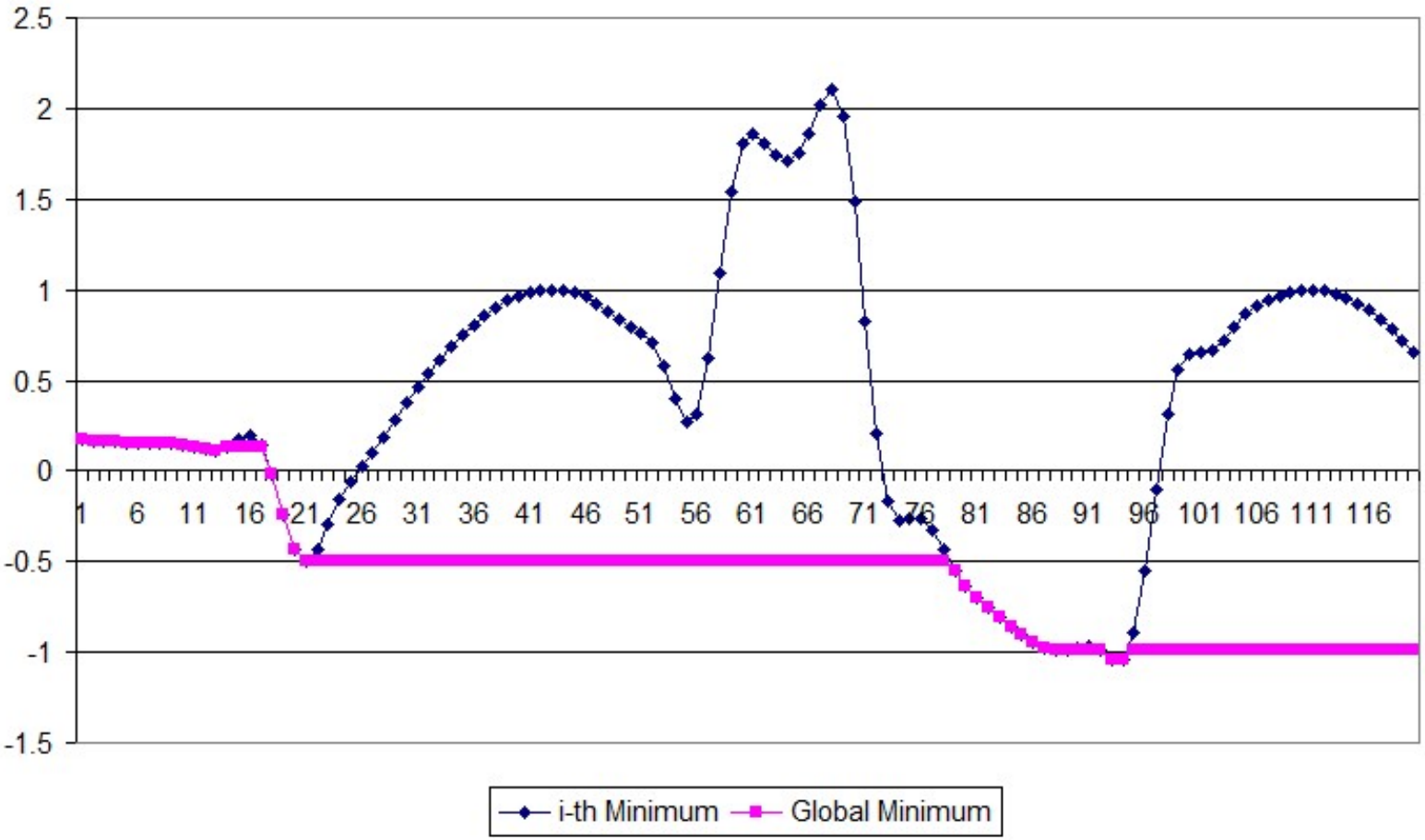
Genetic Algorithms

In his paper "Convergence analysis of canonical genetic algorithms" (see citation), Rudolph states: "*We analyze the convergence properties of the canonical genetic algorithm (CGA) with mutation, crossover and proportional reproduction applied to static optimization problems. It is proved ... that, provided that the best solution in the population is maintained,* ***variants of CGA always converge to the global optimum***"

G. Rudolph, "Convergence analysis of canonical genetic algorithms," in IEEE Transactions on Neural Networks, vol. 5, no. 1, pp. 96-101, Jan. 1994, doi: 10.1109/72.265964.

Genetic Algorithms

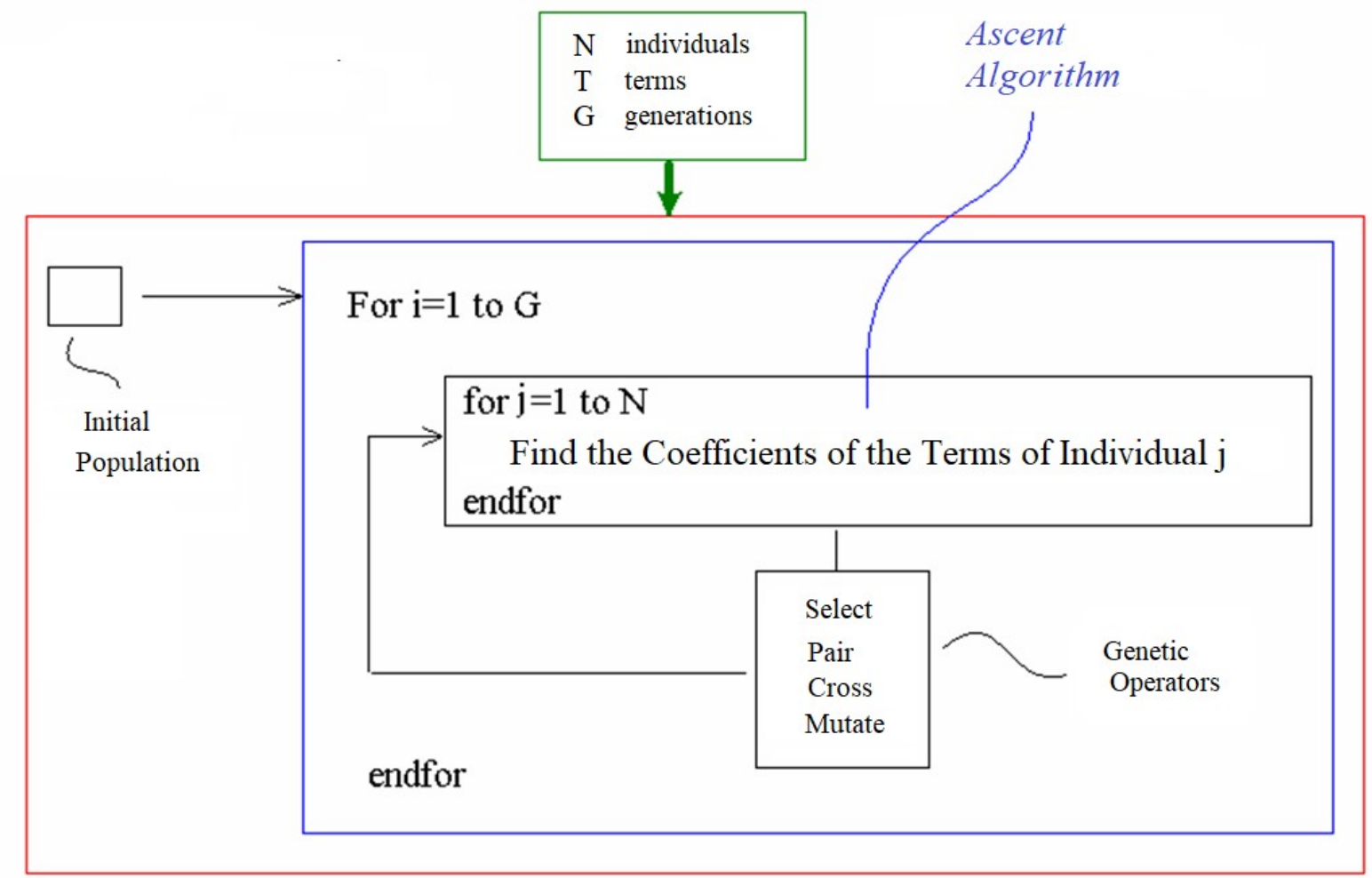
This is a very strong assertion but it may be simply understood from the following graph. (We consider a minimization problem)



Not all Genetic Algorithms are the same

Algorithm	Average Minimum	Relative Efficiency
EGA	0.0635	100.00%
SGA	0.1260	50.43%
RMH	0.1491	42.60%
CHC	0.1501	42.32%
TGA	0.2272	27.96%

Finding the Terms with EGA



EGA and the Ascent Algorithm may be used together to find the coefficients when the specified number of terms (m) has been selected.
Now we would like to know how to specify such an important hyper-parameter.

The last issue we must tackle is:
How do we find the best value of m in

$$Y = c_1 X_1 + c_2 X_2 + \ldots + c_m X_m \ ?$$

Determining the Number of Terms

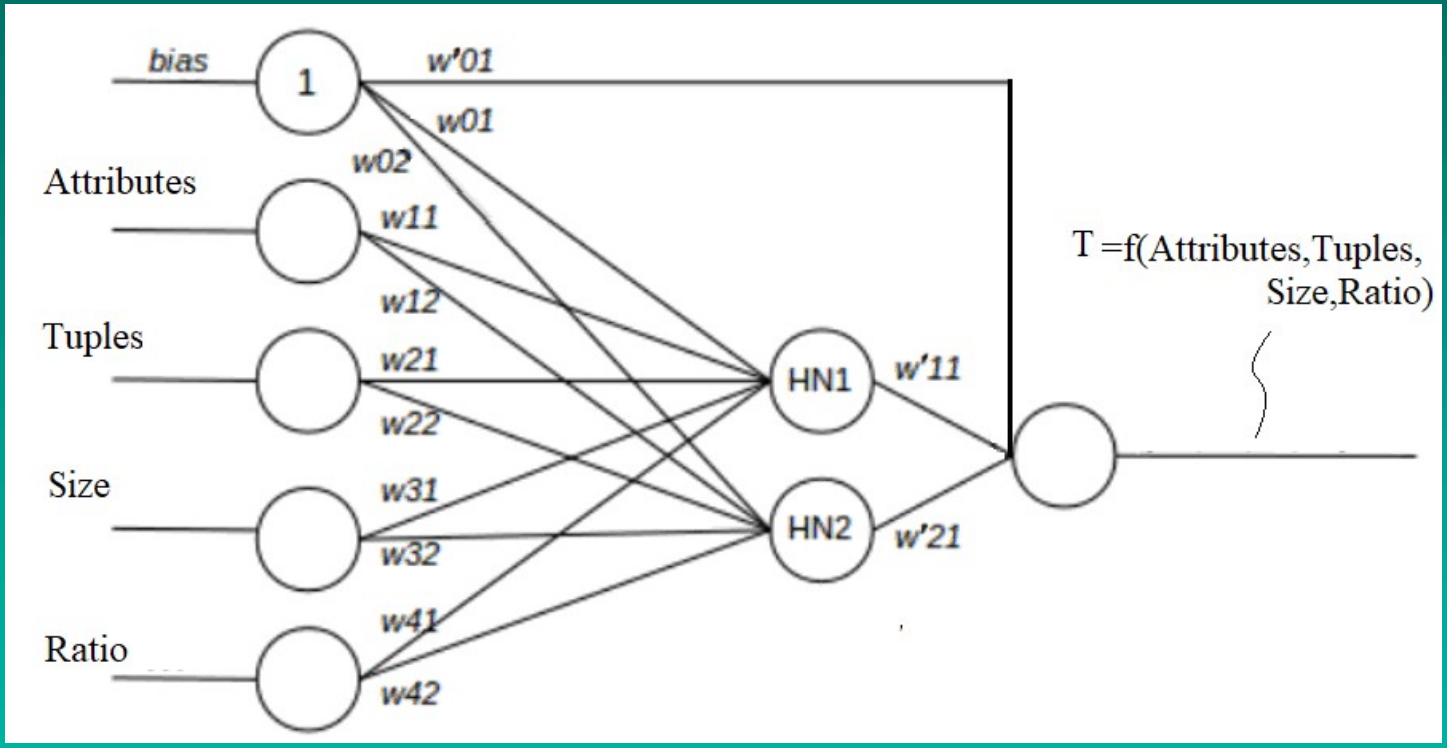
We collected 46 datasets from the University of California Machine Learning dataset repository and the Knowledge Extraction Evolutionary Learning dataset repository. To begin with, 32 of these datasets were chosen and were solved for the multi-variate polynomial EGA. For every $T \in [3,13]$ a polynomial was found and the number of terms corresponding to the best fit was recorded. A total of 352 (11 x 32) polynomials, therefore, were calculated.

With these a neural network (NNt) was trained

Partial Table of the best values of T for the selected datasets

ID	Dataset Name	# Attributes	# Tuples	Size	Comp Size	Comp Ratio	Expected T
1	Breast Cancer wisconsin	10	364	38121	1003	38.0070	8
2	Protein localization sites	7	336	28140	7548	3.7281	7
3	Servomechanism	13	167	22610	771	29.3256	6
4	Yeast	9	1484	140066	32021	4.3742	7
5	Abalone	11	3133	360864	39891	9.0463	6
6	Car Evaluation	22	1728	216384	34636	6.2474	10
7	CPU	36	209	28398	5348	5.3100	9
8	Hepatitis	16	125	30384	5906	5.1446	7
9	Wine	14	125	25986	8876	2.9277	13
10	IRIS	4	150	9120	2685	3.3966	5
11	Facebook	21	500	55200	14521	3.8014	6
12	Whole Sale	10	440	46112	13649	3.3784	8
13	3D road network	2	871	53261	15941	3.3400	10
14	Air quality	8	1200	217522	52252	4.1600	6
15	Air foil self noise	5	1503	182089	43915	4.1500	4
16	Concrete strength	8	1030	18752	52062	3.5900	8
17	Auto mpg	6	398	64368	14798	4.3500	5
18	Credit approval	15	690	215937	42663	5.0600	6
19	Gas turbine propulsion	2	1000	966816	259129	3.7300	5
20	Energy efficiency	11	768	185506	36553	5.0700	9

The Architecture of NNt



A Case of Study

Determination of Wines' Classes Revisited

Machine Learning and Empirical Kowledge

We will use a system for the classification of three wines.

We must find, from a data base, the way in which the class depends on the following 13 variables:

1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10) Color intensity
11) Hue
12) OD280/OD315 of diluted wines
13) Proline

Previous Results

This is a well known data base. It has been tackled with various Learning Machine methods. In this regard the authors remark:

"The data was used with many others for comparing various classifiers. The classes are separable, though only RDA has achieved 100% correct classification.
(RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed data)). All results using the leave-one-out technique."

Machine Learning and Empirical Kowledge

We are provided with data in this way:

Alcohol	Malic	Ash	Alcalinity	Magnesium	Total	Flavanoids	Nonflavano	Proanthocy	Color	Hue	OD280_OD31	Proline	Clase
0.84	0.19	0.57	0.26	0.62	0.63	0.57	0.28	0.59	0.37	0.46	0.97	0.56	0
0.56	0.32	0.70	0.41	0.34	0.63	0.61	0.32	0.76	0.38	0.45	0.70	0.65	0
0.88	0.24	0.61	0.32	0.47	0.99	0.66	0.21	0.56	0.56	0.31	0.80	0.86	0
0.58	0.37	0.81	0.54	0.52	0.63	0.50	0.49	0.44	0.26	0.46	0.61	0.33	0
0.88	0.22	0.58	0.21	0.28	0.52	0.46	0.32	0.50	0.34	0.44	0.85	0.72	0
0.35	0.04	0.00	0.00	0.20	0.34	0.05	0.28	0.00	0.06	0.46	0.20	0.17	0.5
0.34	0.07	0.49	0.28	0.34	0.37	0.16	0.94	0.00	0.17	0.63	0.15	0.29	0.5
0.69	0.10	0.30	0.38	0.26	0.39	0.31	0.36	0.10	0.22	0.61	0.44	0.25	0.5
0.35	0.08	0.43	0.43	0.18	0.87	0.58	0.11	0.46	0.27	0.60	0.59	0.10	0.5
0.30	0.14	0.63	0.43	0.37	0.31	0.30	0.60	0.20	0.14	0.79	0.35	0.05	0.5
0.48	0.12	0.51	0.38	0.57	0.18	0.19	0.15	0.17	0.24	0.23	0.01	0.25	1
0.49	0.44	0.56	0.48	0.37	0.11	0.19	0.21	0.13	0.35	0.21	0.05	0.18	1
0.47	0.31	0.56	0.69	0.30	0.06	0.16	0.26	0.13	0.38	0.15	0.03	0.20	1
0.44	0.56	0.53	0.56	0.39	0.25	0.18	0.08	0.14	0.32	0.24	0.01	0.23	1
0.32	0.79	0.63	0.54	0.21	0.14	0.03	0.75	0.12	0.22	0.22	0.00	0.32	1

Eliminating the “Black Box Machine”

Now we may try to identify an algebraic function eliminating the “black box” disadvantage.

The idea is simple: find the explicit algebraic expression replacing the perceptrons by the mathematical expression of a logistic function.

To do this directly is virtually impossible since *logist(x)* has an infinite expansion.

Finding the Terms with EGA

$$Class = c_1 + c_2 V_{12} + c_3 V_2 + c_4 V_9 + c_5 V_7 + c_6 V_5 + c_7 V_4 V_5^2 + c_8 V_4 V_6 V_7 + c_9 V_4 V_6 V_9 + C_{10} V_9^2 V_{12}$$

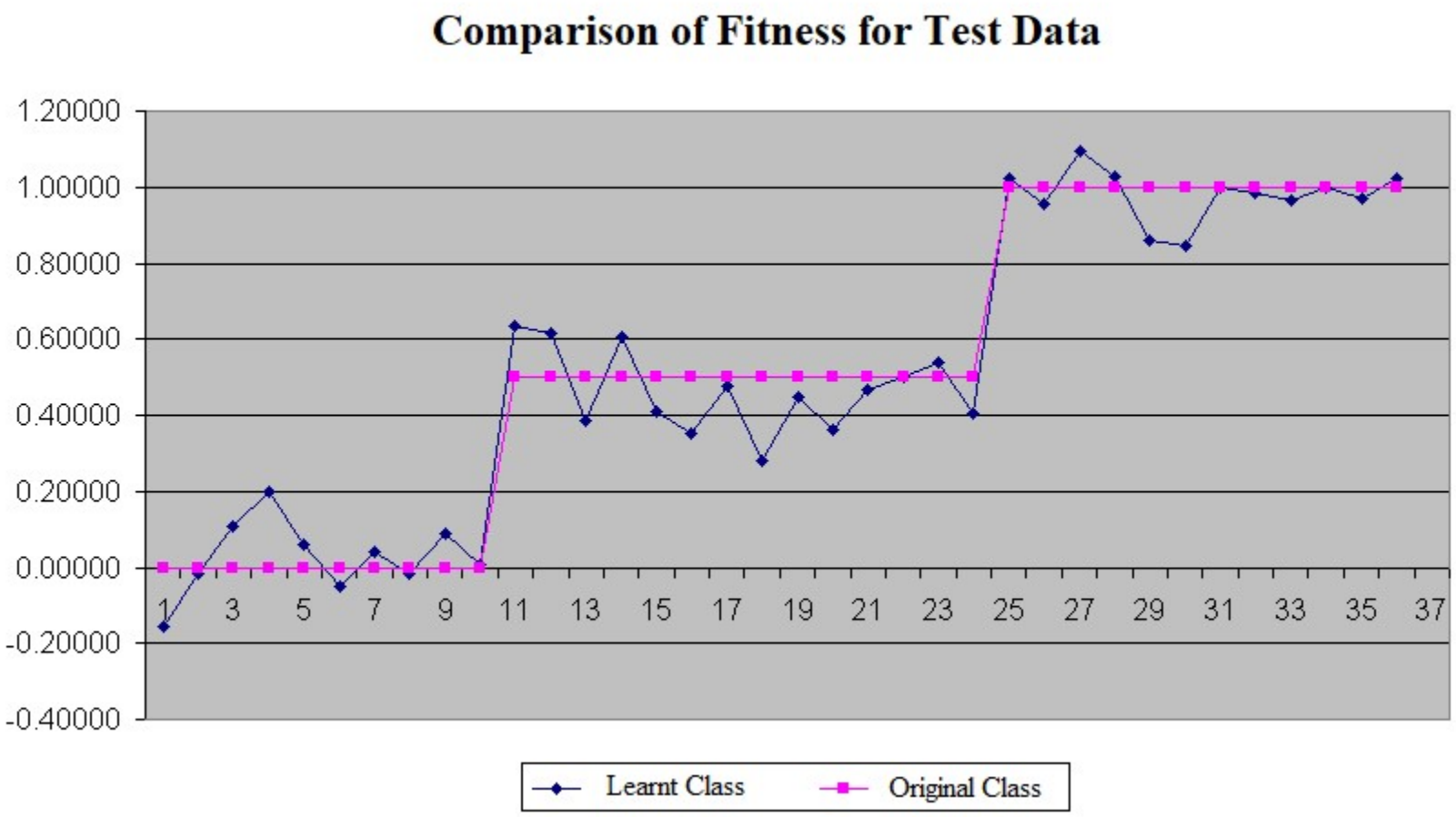
Powers of the variables

Coefficients

C1	C00,00,00,00,00,00,00,00,00,00,00,00,00	1.15786278304368
C2	C00,00,00,00,00,00,00,00,00,00,00,01,00	-0.67433383707615
C3	C00,01,00,00,00,00,00,00,00,00,00,00,00	0.21985118632381
C4	C00,00,00,00,00,00,00,00,01,00,00,00,00	0.46804769903188
C5	C00,00,00,00,00,00,01,00,00,00,00,00,00	-1.44641916040040
C6	C00,00,00,00,01,00,00,00,00,00,00,00,00	-0.79232740379897
C7	C00,00,00,01,02,00,00,00,00,00,00,00,00	1.22274488873102
C8	C00,00,00,01,00,01,01,00,00,00,00,00,00	2.08391092028307
C9	C00,00,00,01,00,01,00,00,01,00,00,00,00	-1.38129259788910
C10	C00,00,00,00,00,00,00,00,02,00,00,01,00	0.34791456797269

1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10) Color intensity
11) Hue
12) OD280/OD315 of diluted wines
13) Proline

Finding the Class Equation with EGA



The Multivariate Algebraic Equation is has been Found

1. The polynomial accepts (and returns) real values, whereas the classes are denoted by integers. In this regard it is interesting to note that given the values assigned to the classes used to train the EGA (Class 1=0.0; Class 2=0.5; Class 3=1.0) any value between 0 and 0.25 maps into class 1, any value between 0.25 and 0.75 maps into class 2 and any value >0.75 maps into class 3
2. Before we noted that according to previous reported results "Only RDA has achieved 100% correct classification. (RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 96.1% (z-transformed data))". Here 100% accuracy was achieved on TEST DATA.

Qualitative Interpretation of the Multivariate Equation is now Possible

Right away we see that variables V_1, V_3, V_8, V_{10}, V_{11} and V_{13} do not appear in the class equation. This means that Alcohol, Ash, Nonflavanoid phenols, Color intensity, Hue and Proline are irrelavant for the classification purposes.

This is remarkable given that the 5 of the 13 selected attributes by the experts are useless for classification purposes.

From the Class Equation

- OD280/OD315 of diluted wines, Malic Acid, Proanthocyanins, Flavanoids and Magnesium are all uncorrelated
- Alcalinity of ash is correlated to the squared value of Magnesium
- Alcalinity of ash, Total phenols and Flavanoids have a linear relationship
- Alcalinity of ash and Total phenols have a linear relationship
- The square of Proanthocyanins is related to OD280/OD315 of diluted wines

From the Class Equation

- Phenols is the most affine
- Proanthocyanins is as affine as Phenols
- The relations between the variables are never of degree higher than 3
- The largest relative affinity is the combination of Alcalinity, Phenols and Flavanoids

From the Class Equation

As may be seen, the multivariate approximation polynomial may be as effective as older techniques

It allows us to derive general characteristics as to how a variable and/or set of variables affects the classes