

Harmonic Theory and Machine Learning

Jorge Nanclares & Ulises M. A. Rapallini
 Universidad Tecnológica Nacional
 Facultad Regional Concepción del Uruguay
 GEINAR - Grupo de Estudio en Inteligencia Artificial
 Entre Rios, Argentina
 nanclarj@frcu,utn.edu.ar, rapalliniu@frcu.utn.edu.ar

June 4, 2007

Abstract

A natural inference mechanism is presented : the Black Box problem is transformed into a Dirichlet's problem on the closed cube. Then it is solved in closed polynomial form, together with a Mean-Value theorem and a Maximum Principle. A generalization to Polytopes and a reduction of any Dirichlet problem on compacta is mapped into a unit cube in more dimensions. An algorithm for calculating the solution is suggested.

keywords : Neural Network, Machine Learning, Potential Theory, Polynomial Approximation

1 HEURISTIC INTRODUCTION

One of the main questions facing the field of Artificial Intelligence is the following input/output problem : Given a finite number m of instances or cases of a function as training data :

$$f : \{0, 1\}^n \rightarrow \{0, 1\}$$

infer , relying only on the given training data, unknown cases or instances of f , generalizing or predicting those yet unknown cases. For example, we are given cases of a binary function :

$$\begin{aligned} 0, 1, 1, 0, 0, 0, 0 &\rightarrow 0 \\ 1, 1, 1, 0, 0, 1, 0 &\rightarrow 0 \\ 0, 0, 0, 1, 1, 1, 1 &\rightarrow 1 \end{aligned}$$

and we have to predict , say :

$$0, 1, 0, 0, 0, 0, 1 \rightarrow ?$$

The problem may be restated as follows :

Given a binary function on a subset of the vertices of the n th dimensional unit cube, infer its values on the rest of the vertices. We must find a way of extending the given data without introducing extra information. The flow of heat is a powerful process for flattening boundary values, losing information until a steady state is reached with a final minimum of potential energy and a maximum of entropy (in short, with all possible flatness).

The value of a harmonic function in the center of a ball equals the average of its values on the boundary sphere ; so at every point any sharpness of the harmonic function is lost, flattened out by averaging. Not only do the maxima and minima occur at the boundary but all information present at the boundary is lost inside the domain in the process of temperature diffusion, once a steady state is attained in the limit smoothed harmonic function. (The Boltzman entropy of the initial distribution of temperature may be calculated as $k \log(P)$ where k is the Boltzmann constant and P is the number of microstates compatible with the actual macrostate, that is, with the actual value of our function at the boundary). Thus we may say heuristically that the information of the whole domain D with its boundary , the training data and the final harmonic function equals the

information at the boundary when the process starts, the rest is a powerful flattening process ending in a function without local maxima or minima. Thus the process adds no information to the initial training data.

Consider the n-dimensional temperature flow (heat equation) :

$$\frac{\partial T(X_1, \dots, X_n, t)}{\partial t} = \sum \frac{\partial^2 T(X_1, \dots, X_n, t)}{\partial X_i^2}$$

such that on the vertices of the cube in R^n a function $T(X_1, \dots, X_n, 0)$ is given as an initial temperature distribution .We only fix T as given data at some vertices of the cube. If we let T flow to the rest of the cube, always maintaining T at the chosen vertices, and let $t \rightarrow \infty$,once the steady sate (not equilibrium) is reached T will take values on the whole cube and we may predict the values of T on all the vertices. The limit function will be harmonic .

The most important property of the solution is this : the potential equation solves the following variational problem :

In a given physical system find the C_2 function f on the n unit cube compatible with the given data and with minimum potential energy :

$$E = \min_f \int_0^1 \dots \int_0^1 \sqrt{\sum_{i=1}^n \left(\frac{\partial f}{\partial X_i}\right)^2} dX_1 \dots dX_n$$

2 ON THE UNIT CUBE

Definition :

A P1P (Potential polynomial of degree 1 in each variable) will consist of a finite sum of monomials in the variables :

$$X_1, \dots, X_n$$

such that each (real valued) variable has at most degree 1; all the coefficients also are real-valued.

The most general P1P with real-valued coefficients will have the form :

$$c + \sum c_i X_i + \sum c_{ij} X_i X_j + \sum c_{ijk} X_i X_j X_k + \dots + c_{1.2\dots n} X_1 X_2 \dots X_n$$

where all the sums run over all the different sub-indices i, j, \dots , and where no repeated sub-indices are allowed (there is no $c_{\dots j, \dots, j, \dots}$). A special case is the purely boolean where the monomials are composed with variables X_i or their boolean negations $1 - X_j$ and the coefficients may be 0 or 1.

Every well formed formula in the predicate calculus has a logical equivalent in this disjunctive normal form .

P1Ps :

$$P(X_1, X_2, \dots, X_n)$$

are functions in the space R^n : their derivatives

$$\frac{\partial P}{\partial X_i} = Q(X_1, X_2, \dots, X_n)$$

also are P1Ps, independent of X_i , which implies that all second derivatives are zero , which makes P1Ps harmonic functions in R^n .

Consider again the n-dimensional heat equation :

$$\frac{\partial T(X_1, \dots, X_n, t)}{\partial t} = \sum \frac{\partial^2 T(X_1, \dots, X_n, t)}{\partial X_i^2}$$

On the vertices of the cube in R^n a function $T(X_1, \dots, X_n, 0)$ is given as a fixed initial temperature distribution .

Stepwise we may let it flow until the system converges to a steady state , keeping the initial data fixed, first to edges ,once the edges attain a steady state which will be preserved , then to 2-dimensional facets, and so on until we have a temperature distribution on all the boundary such that it preserves the given data with un ceasing flow of heat at the initial vertices which

preserves the initial information, and last allowing it to diffuse into the interior points of the n-dimensional cube.

The resulting function when $t \rightarrow \infty$ is the unique solution of Dirichlet's problem (see ref.1 and 4) :

Keeping boundary data on some vertices fixed (it may be just 1 vertex) find in the whole cube the solution of Laplace's equation :

$$\sum \frac{\partial^2 T(X_1, \dots, X_n)}{\partial X_i^2} = 0$$

This final steady state is not the state of equilibrium (ref.10) . The flow of heat in order to sustain the boundary conditions keeps the system away from it.

We are given a finite number of instances or cases of a function f on a subset of the vertices of the cube in R^n , and we are required to find f on all the cube and to express it as a P1P on the training data Let :

$$X_1^k, X_2^k, \dots, X_n^k (k = 1, \dots, m)$$

be boolean variables, identified with m vertices of the n-dimensional cube, and let

$$f(X_1^k, X_2^k, \dots, X_n^k)$$

the m given values of f as training data. Call V the convex hull of the m vertices.

1. Assign to the edge of two linked vertices a P1P consistent with the values of f on the linked vertices .That is, assign 0 or 1 or some X_i or some $1 - X_j$ to the actual edge, according to the training data .

2. Proceed in the same way with all the facets of V , stepwise on each and all dimensions of the boundary of V . In each case we are solving Laplace's

equation with P1Ps on each boundary of each facet of ∂V .

3. When all the boundary of V has been modelled, again solve Laplace's equation, this last time in n dimensions, to get a unique harmonic function on V , again a P1P in n variables.

4. The expression of the P1P thus obtained is automatically extended with identical expression, to all the cube (in fact to all R^n).

If we assume there is data on a few points of the closed cube, then any continuous function defined at those points might grow to be a solution to Laplace's equation inside the cube (ref.1). We get unicity through 'our'

Dirichlet problem :

The following is our main result :

Theorem 1

There is only one solution f for our Dirichlet problem on the closed cube taking the f_i values on its vertices, which is harmonic on each & all the facets of the cube.

Proof : The case n=0 is obvious. Consider a matrix M_k of order 2^k whose rows stand for the different vertices of the cube, and whose columns are :

$$\begin{aligned} &1, X_1, X_2, \dots, X_n, X_1 X_2, \dots, \\ &X_{n-1} X_n, X_1 X_2 X_3, \dots, \\ &X_1 X_2 \dots X_k \end{aligned}$$

We choose the first row or vertex to be $\{1,0,0,\dots,0\}$, the case in which the variables are zero; and the first column to be $\{1,1,\dots,1\}$ corresponding to 1. M_k is a matrix of order 2^k . For

example if $n = 1$ we have M_1 is :

$$\begin{matrix} 1 & 0 \\ 1 & 1 \end{matrix}$$

and M_2 is :

$$\begin{matrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{matrix}$$

We proceed to generate M_{k+1} : the columns are the previous ones plus the new ones:

$$X_{k+1} * [1, X_1, \dots, X_1 X_2 \dots X_k]$$

We may write :

$$M_{k+1} = \begin{pmatrix} M_k & 0 * M_k \\ M_k & 1 * M_k \end{pmatrix} = \begin{pmatrix} M_k & 0 \\ M_k & M_k \end{pmatrix}$$

An elementary exercise in algebra gives us the result : $det(M_{k+1}) = det(M_k)^2$.

By the induction hypothesis the determinant of M_k is not zero, then the determinant of M_{k+1} is not zero .

The linear equations :

$$M_n \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{2^n} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{2^n} \end{pmatrix}$$

for any given f_i have exactly one solution; then :

$$f = \lambda_1 + \lambda_2 X_1 + \dots + \lambda_{2^n} X_1 X_2 \dots X_n$$

is the unique solution to our Dirichlet problem on the closed n-dimensional unit cube.

3 MODELLING WITH P1P'S

The potential energy of each function f is calculated from the variational formula :

$$E = \min_f \int_0^1 \dots \int_0^1 \sqrt{\sum_{i=1}^n \left(\frac{\partial f}{\partial X_i}\right)^2} dX_1 \dots dX_n$$

The following data are a modified form of a decision tree solving a 'tennis puzzle' (ref. 15, chapter 2). A Boolean function fis defined in 13 examples :

0 0 1 1 1 0 → 0
 0 0 1 1 1 1 → 0
 0 1 1 1 1 0 → 1
 1 1 0 0 1 0 → 1
 1 1 1 0 0 0 → 1
 1 1 0 1 0 1 → 0
 0 0 0 0 0 1 → 1
 0 0 0 0 1 0 → 0
 0 1 0 0 0 0 → 1
 1 1 0 0 0 0 → 1
 0 1 0 0 1 1 → 1
 0 1 1 1 0 0 → 1
 1 1 0 0 1 1 → 0

We proceed to calculate a least squares linear fit to the data; all the 2^6 regressors are used. We find the P1P model :

$$1 - X_5 + X_2 X_5 - X_1 X_2 X_6$$

as a fairly good representation of the function.

Warning :

Our solution does not always end in a minimum description length P1P.

If we are given :

0 0 — → 0
 1 0 — → 1
 0 1 — → 0

and we are asked to find the value corresponding to 1 1 :

The flow of heat will give us the chosen solution :

1 1 — → 0.5

model =

$$X_2(1 - 0.5X_1)$$

with potential energy = 1 .354

If instead 0.5 we put :

1 1 — → 0

we get a potential energy = 1.367 and a model =

$$X_1(1 - X_2)$$

But with :

$$1\ 1 \rightarrow 1$$

we get a potential energy = 1.4948 and a model =

$$X_1$$

In other words : the solution model is the longest.

Again, given :

$$\begin{aligned} 0\ 0 &\rightarrow 0 \\ 1\ 0 &\rightarrow 1 \\ 1\ 1 &\rightarrow 0 \end{aligned}$$

The flow of heat will give us :

$$0\ 1 \rightarrow 0$$

with potential energy = 1.367 and model =

$$X_1(1 - X_2)$$

If we put:

$$0\ 1 \rightarrow 1$$

we get a potential energy = 1.416 and a model =

$$X_1(1 - X_2) + (1 - X_1)X_2$$

If we put :

$$0\ 1 \rightarrow -1$$

the potential energy = 1.416 and the model =

$$X_1X_2$$

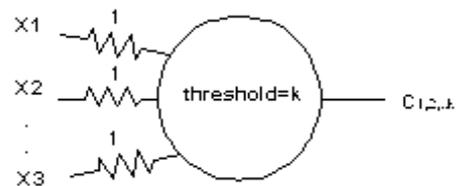
The first solution is chosen, and it is not the shortest one.

4 NEURAL NETS

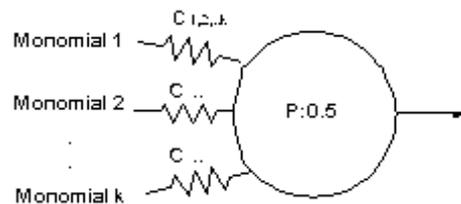
Every P1P has an equivalent Neural Net : Each variable X_i is an input neuron and the rest is an and/or scheme ; the real coefficient of each monomial is the weight of the corresponding neuron. For example :

$$P = c + \sum c_i X_i + \sum c_{ij} X_i X_j + \dots + c_{12\dots n} X_1 X_2 \dots X_n$$

may be directly interpreted as a neural net and as a NeuPro code. weights are set to 1 and the threshold to k . The monomial will enter other neurons with a weight equal to its coefficient c .

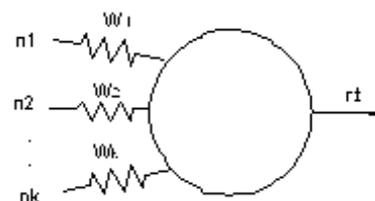


On the other hand all the monomial neurons will be connected to neuron P with a threshold equal to 0.5.



The weight of each monomial is its coefficient in the P1P.

Consider as example :



It translates directly into the following NeuPro code (see ref.12) :

$$r : t :- n1 : w1, n2 : w2, \dots, nk : wk.$$

We started with an I/O situation from which we may infer the underlying function f expressing it as a P1P ; then we obtain a 'natural' neural net representing f and finally we translate mechanically the neural net into a NeuPro program.

5 CONCLUDING REMARKS

1. Assuming training data is given at a subset of the vertices of a unit n cube we map the prediction problem into finding the solution of the heat equation in the whole cube.

2. Our inference machine is then the natural flow of temperature fixing the training data during the process in a typical Dissipative Structure. (see ref.10)

3. The unique limit solution as $t \rightarrow \infty$ is harmonic which means it is the function with minimum potential energy compatible with the given data.

4. The solution also has maximum Boltzmann's entropy :

$$H = k \log(P)$$

where P is the number of microstates (or complexions) in the actual physical final macrostate and k is Boltzmann's constant.

5. The P1P solution has an immediate translation into a Neural Net and into a Neupro (or Prolog) code.

6. We may translate Prolog code into P1P's obtaining a model of the code, a kind of 'self model' of the program. (see appendix). This feature might be useful in the debugging process.

7. The P1P solution is often, but not always, a Minimum Description Length object.

8. A forthcoming paper will study the complexity properties of our scheme.

6 APPENDIX

We map into our scheme a prolog program modelling the grandfather relation . Consider the program :

$$a(X, Z) :- p(X, Y), p(Y, Z).$$

p(juan, pedro).

p(juan, luis).

p(jose, jorge).

p(pedro, alberto).

p(jorge, roberto).

a(juan, alberto).

a(jose, roberto).

We may code this information as follows:

We assign predicates p and a with 1 and 0 respectively.

juan=000

pedro=001

luis=010

jose=011

jorge=100

alberto=101

roberto=110

ruben=111

We code the predicates as follows:

1110001 is the code for p(ruben, juan).

0000011 is the code for p(juan, pedro).

0000101 is the code for p(juan, luis).

0111001 is the code for p(jose, jorge).

0011011 is the code for p(pedro, alberto).

1001101 is the code for p(jorge, roberto).

0001010 is the code for a(juan, alberto).

0111100 is the code for a(jose, roberto).

We find, as a fitted model for the underlying theory with an error = 2.4^{-15}

the following function :

$$f = 2 - X_2 + X_3 - X_2X_3 - X_5 - X_6 - X_4X_6$$

Now we test this model for f trying the following examples :

1110010 is the code for a(ruben ,pedro)
(f should be 0) .

1110100 is the code for a(ruben ,luis)
(f should be 0) .

1000010 is the code for a(jorge ,pedro)
(f should be 1) .

We run the model for f and find :

$$\begin{aligned} \{X_1X_2X_3X_4X_5X_6\} &= \\ \{111001\} &\rightarrow f = 2.2^{-15} \\ \{X_1X_2X_3X_4X_5X_6\} &= \\ \{111010\} &\rightarrow f = 2.4^{-15} \\ \{X_1X_2X_3X_4X_5X_6\} &= \\ \{100001\} &\rightarrow f = 1 \end{aligned}$$

In almost perfect agreement with the theory.

7 BIBLIOGRAPHY

References

- [1] Axler S. Bourdon P. W.Ramey : Harmonic Function Theory. Springer Verlag New York 2001.
- [2] Bundy A. : The Computer Modelling of Mathematical Reasoning . Academic Press 1983.
- [3] Chaitin G.:A theory of program size formally identical to Information Theory. Journal ACM vol.22 num.3 July 1975.
- [4] Courant R. & Hilbert D. : Methods of Mathematical Physics. Interscience publishers 1953.
- [5] Hand D. Mannila H. Smyth P. : Principles of Datamining . M.I.T Press 2001.
- [6] Kelley J. : General Topology. Van Nostrand University series in higher mathematics 1955.
- [7] Kon M.A. & Plaskota L.: Information Complexity of Neural Networks. Neural Networks 2000.
- [8] Mashkar H.N. : Neural Networks & Approximation Theory. Neural Networks 9 721-722.
- [9] Mendelson E. : Introduction to Mathematical Logic. Van Nostrand 1964.
- [10] Nicolis G.&Prigogine I. : Exploring Complexity. Freeman & Co New York 1989.
- [11] Poggio T. & Girossi F. : Regularization Algorithms for learning that are equivalent to multilayer Networks. Science 247 978-982.
- [12] Rapallini U. & Nanclares J. : Intérprete NeuPro utilizando la NeuPro Abstract Machine. XI Congreso Argentino de Ciencias de la Computación. Concordia Entre Ríos octubre de 2005.
- [13] Shannon C.: A Mathematical Theory of Communication. Univ. of Illinois Press 1949.
- [14] Sussman H. : Semigroup representations bilinear approximation of input/output maps and generalized inputs. Lecture notes in Economics & Math.Systems. Mathematical Systems Theory 131 Udine 1975 Springer Verlag.
- [15] Tom Mitchel : Machine Learning. WCB/McGraw Hill 1997.
- [16] Traub J.F.&Werschultz :Complexity & Information Cambridge University Press 1999 Cambridge.