REVISTA DE LA
UNIÓN MATEMÁTICA ARGENTINA
Volumen 48, Número 1, 2007, Páginas 65–74

presented by R. A. Macías

HARMONIC FUNCTIONS ON THE CLOSED CUBE: AN APPLICATION TO LEARNING THEORY

O.R. FAURE, J. NANCLARES, AND U. RAPALLINI

Abstract. A natural inference mechanism is presented: the Black Box problem is transformed into a Dirichlet problem on the closed cube. Then it is solved in closed polynomial form, together with a Mean-Value Theorem and a Maximum Principle. An algorithm for calculating the solution is suggested. A special feedforward neural net is deducted for each polynomial.

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:

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

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:

$$\begin{array}{lcl} (0,1,1,0,0,0,0) & \longrightarrow & 0 \\ (0,1,1,0,0,0,0) & \longrightarrow & 0 \\ (0,0,0,1,1,1,1) & \longrightarrow & 1 \end{array}$$

and we have to predict, say:

$$(0,1,0,0,0,0,1) \quad \longrightarrow \quad ?$$

The problem may be restated as follows:

Given a binary function on a subset of the vertices of the nth 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.

2000 *Mathematics Subject Classification.* 31-99, 52-99.
Key words and phrases. Dirichlet Problem, Diffusion Equation, Black Box, Convex Bodies.

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).

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 $\mathbb{R}^n$ 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 $t \to \infty$, 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 $\mathcal{C}_2$ 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 2.1. *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:

$$c + \sum c_i X_i + \sum c_{i,j} X_i X_j + \sum c_{i,j,k} X_i X_j X_k \ + \dots + c_{1,2,\dots,n} X_1 X_2 \dots X_n$$

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.

Every well formed formula in the predicate calculus has a logical equivalent in this disjuntive normal form. P1Ps:

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

are functions in the space $\mathbb{R}^n$: their derivatives

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

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

Consider again the n-dimensional heat equation:

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

On the vertices of the cube in $\mathbb{R}^{\mathrm{n}}$ a function $T(X_1, \ldots, X_n, 0)$ 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 unceasing 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 $t \to \infty$ is the unique solution of Dirichlet's problem (see [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:

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

This final steady state is not the state of equilibrium (see [9]). 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 $\mathbb{R}^n$, and we are required to find f on all the cube and to express it as a P1P on the training data.

Let:

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

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

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

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 $P1P$s 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 $\mathbb{R}^n$). 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 (see [1]).

We get unicity through 'our' Dirichlet problem.

The following is our main result:

Theorem 2.1. *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:

$$1, X_1, X_{2,} \ldots, X_n, X_1X_2, \ldots, X_{n-1}X_n, X_1X_2X_3, \ldots, X_1X_2 \ldots X_k$$

We choose the first row or vertex to be $\{1, 0, 0, \ldots, 0\}$, the case in which the variables are zero; and the first columnto be $\{1, 1, \ldots, 1\}$ corresponding to 1. M_k is a matrix of order 2^k. For example if $n = 1$ we have M_1 is:

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

and M_2 is:

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

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

$$X_{k+1} * \ [\ 1, X_1, \ldots, X_1X_2 \ldots X_k]$$

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 excercise 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:

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

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

3. Modelling P1P's

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

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

The following data are a modified form of a decision tree solving a 'tennis puzzle'(see [14], Chapter 2).

A Boolean function f is defined in 13 examples:

$$\begin{array}{lcl}
(0,0,1,1,1,0) & \longrightarrow & 0 \\
(0,0,1,1,1,1) & \longrightarrow & 0 \\
(0,1,1,1,1,0) & \longrightarrow & 1 \\
(1,1,0,0,1,0) & \longrightarrow & 1 \\
(1,1,1,0,0,0) & \longrightarrow & 1 \\
(1,1,0,1,0,1) & \longrightarrow & 0 \\
(0,0,0,0,0,1) & \longrightarrow & 1 \\
(0,0,0,0,1,0) & \longrightarrow & 0 \\
(0,1,0,0,0,0) & \longrightarrow & 1 \\
(1,1,0,0,0,0) & \longrightarrow & 1 \\
(0,1,0,0,1,1) & \longrightarrow & 1 \\
(0,1,1,1,0,0) & \longrightarrow & 1 \\
(1,1,0,0,1,1) & \longrightarrow & 0
\end{array}$$

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 :

$$\begin{array}{ccc}(0,0) & \longrightarrow & 0\\ (1,0) & \longrightarrow & 1\\ (0,1) & \longrightarrow & 0\end{array}$$

and we are asked to find the value corresponding to $(1,1)$:

The flow of heat will give us the chosen solution:

$$(0,0) \quad \longrightarrow \quad 0.5$$

with the following model

$$X_2(1-0.5X_1)$$

with potential energy equal to 1.354.

If instead 0.5 we put:

$$(1,1) \quad \longrightarrow \quad 0$$

we get a potential energy equal to 1.367 and a model given by $X_1(1-X_2)$ But with:

$$(1,1) \quad \longrightarrow \quad 1$$

we get a potential energy equal to 1.4948 and a model given by X_1.

In other words: the solution model is the longest.

Again, given :

$$\begin{array}{ccc}(0,0) & \longrightarrow & 0\\ (1,0) & \longrightarrow & 1\\ (1,1) & \longrightarrow & 0\end{array}$$

The flow of heat will give us:

$$(0,1) \quad \longrightarrow \quad 0$$

with potential energy 1.367 and model given by $X_1(1-X_2)$.

If we put:

$$(0,1) \quad \longrightarrow \quad 1$$

we get a potential energy equal to 1.416 and a model is

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

If we put :

$$(0,1) \quad \longrightarrow \quad -1$$

the potential energy is equal 1.416 and the model is X_1-X_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:

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

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.

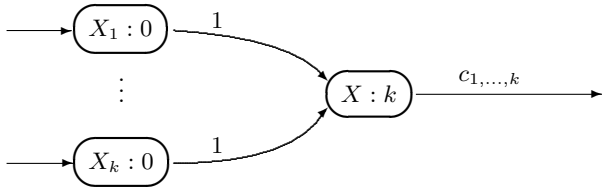

Figure 1. Monomial: $X = c_{1,\ldots,k} X_1 X_2 \ldots X_k$

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

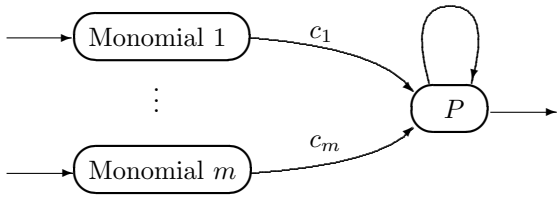

Figure 2. $P = c + c_1$ Monomial $1 + \cdots + c_m$ Monomial m

The weight of each monomial is its coefficient in the P1P. Consider as example:

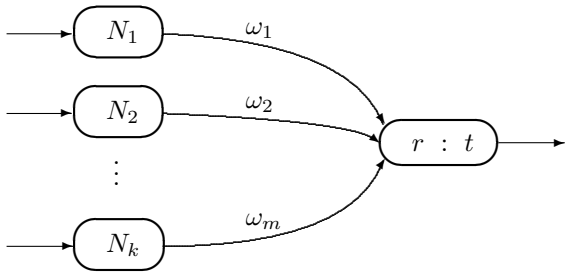

Figure 3.

It translates directly into the following NeuPro code (see [11]):

$$(r:t) \quad :- \quad (n_1:\omega_1),\ \ (n_2:\omega_2), \quad \ldots \quad ,(n_k:\omega_k)$$

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 [9])
(3) The unique limit solution as $t \to \infty$ 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 inmediate translation into a neural net and into a Neupro (or Prolog) code.
(6) We may translate Prolog code into P1P's obtaning 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.

6. Appendix

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

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

$p(\text{Juan}, \text{Pedro}).$
$p(\text{Juan}, \text{Luis}).$
$p(\text{Jose}, \text{Jorge}).$
$p(\text{Pedro}, \text{Alberto}).$
$p(\text{Jorge}, \text{Roberto}).$

$a(\text{Juan}, \text{Alberto}).$
$a(\text{Jose}, \text{Roberto}).$

Its meaning: relation a holds between objects X and Z if $(:-)$ relation p holds between objects X and Y and $(,\ )$ relation p holds between Y and Z.

We may code this information as follows: We assign predicates p and a with 1 and 0 respectively.

$$\begin{array}{lcl} \text{Juan} & = & 000 \\ \text{Pedro} & = & 001 \\ \text{Luis} & = & 010 \\ \text{Jose} & = & 011 \\ \text{Jorge} & = & 100 \\ \text{Alberto} & = & 101 \\ \text{Roberto} & = & 110 \\ \text{Ruben} & = & 111 \end{array}$$

We code the predicates as follows :

$$\begin{array}{lll} (1,1,1,0,0,0,1) & \text{is the code for} & p(\text{Ruben}, \text{Juan}). \\ (0,0,0,0,0,1,1) & \text{is the code for} & p(\text{Juan}, \text{Pedro}). \\ (0,0,0,0,1,0,1) & \text{is the code for} & p(\text{Juan}, \text{Luis}). \\ (0,1,1,1,0,0,1) & \text{is the code for} & p(\text{Jose}, \text{Jorge}). \\ (0,0,1,1,0,1,1) & \text{is the code for} & p(\text{Pedro}, \text{Alberto}). \\ (1,0,0,1,1,0,1) & \text{is the code for} & p(\text{Jorge}, \text{Roberto}). \\ (0,0,0,1,0,1,0) & \text{is the code for} & a(\text{Juan}, \text{Alberto}). \\ (0,1,1,1,1,0,0) & \text{is the code for} & a(\text{Jose}, \text{Roberto}). \end{array}$$

We find, as a fitted model for the underlyng 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:

$$\begin{array}{llll} (1,1,1,0,0,1,0) & \text{is the code for} & a(\text{Ruben}, \text{Pedro}) & (f \text{ should be } 0). \\ (1,1,1,0,1,0,0) & \text{is the code for} & a(\text{Ruben}, \text{Luis}) & (f \text{ should be } 0). \\ (1,0,0,0,0,1,0) & \text{is the code for} & a(\text{Jorge}, \text{Pedro}) & (f \text{ should be } 1). \end{array}$$

We run the model for f and find :

$$\begin{array}{lcl} (X_1, X_2, X_3, X_4, X_5, X_6) & = & (1,1,1,0,0,1) \longrightarrow f = 2.2^{-15} \\ (X_1, X_2, X_3, X_4, X_5, X_6) & = & (1,1,1,0,1,0) \longrightarrow f = 2.4^{-15} \\ (X_1, X_2, X_3, X_4, X_5, X_6) & = & (1,0,0,0,0,1) \longrightarrow f = 1 \end{array}$$

in almost perfect agreement with the theory.

References

[1] Axler S., Bourdon P. & Ramey W. (2001) Harmonic Function Theory. Springer Verlag.

[2] Bundy A. (1983) The Computer Modelling of Mathematical Reasoning. Academic Press.

[3] Chaitin G. (1975) A theory of program size formally identical to Information Theory. Journal ACM Vol. 22 #**3**, 329-340.

[4] Courant R. & Hilbert D. (1953) Methods of Mathematical Physics. Interscience Publishers.

[5] Hand D., Mannila H. & Smyth P. (2001) Principles of Data Mining. MIT Press.

[6] Kelley J. (1955) General Topology. Van Nostrand University Series in Higher Mathematics.

[7] Kon M.A. & Plaskota L. (2000) Information Complexity of Neural Networks. Neural Networks.

[8] Mendelson E. (1964) Introduction to Mathematical Logic. Van Nostrand.

[9] Nicolis G. & Prigogine I. (1989) Exploring Complexity. Freeman & Co.

[10] Poggio T. & Girossi F. (1990) Regularization Algorithms for learning that are equivalent to multilayer Networks. Science # **247** pp. 978-982.

[11] Rapallini U. & Nanclares J. (2005) Intérprete NeuPro utilizando la NeuPro Abstract Machine. XI Congreso Argentino de Ciencias de la Computación. Concordia ER.

[12] Shannon C. (1949) A Mathematical Theory of Communication. University of Illinois Press.

[13] Sussman H. (1975) Semigroup Representations Bilinear Approximation of input/output maps and generalized inputs. Lecture Notes in Economics & Math. Systems. Mathematical Systems Theory 131, Springer Verlag.

[14] Mitchel T. (1997) Machine Learning. WCB McGraw Hill.

[15] Traub J.F. & Werschultz (1999) Complexity & Information. Cambridge University Press.

O. R. Faure
Facultad Regional Concepción del Uruguay,
Universidad Tecnológica Nacional,
Ing. Pereira 676,
E3264BTD Concepción del Uruguay (ER), Argentina
ofaure@frcu.utn.edu.ar

J. Nanclares
Facultad Regional Concepción del Uruguay,
Universidad Tecnológica Nacional,
Ing. Pereira 676,
E3264BTD Concepción del Uruguay (ER), Argentina
nanclarj@frcu.utn.edu.ar

U. Rapallini
Facultad Regional Concepción del Uruguay,
Universidad Tecnológica Nacional,
Ing. Pereira 676,
E3264BTD Concepción del Uruguay (ER), Argentina
rapa@frcu.utn.edu.ar

Recibido: 2 de octubre de 2007
Aceptado: 22 de octubre de 2007