■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

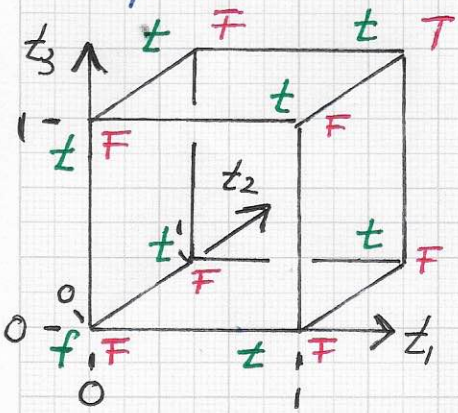
One can possibly employ such a condition ($s_j \approx L_j^{N_j}$), where the number of given data samples for class j defines the class-j-specific values of resolution levels (L_j) and dimensions (N_j). One has to use "practical experimentation" – training, testing, evaluation etc. – to determine near-optimal values for these parameters, to ensure that a required classification performance threshold is statistically met.

• **Note.** The following ideas concerning Boolean algebra / functions, Boolean operators, truth tables etc. have been discussed, to some degree, previously. The ideas focus on the operators PLUS (+), TIMES (·) and OR (∨), AND (∧). For example, one could consider and explore "relationships" between the expressions $B_3(b_1, b_2) = \sum_j \sum_i c_{i,j} H_i(b_1) \cdot H_j(b_2)$ and $\bigvee_j \bigvee_i c_{i,j} \left(T_i(t_1) \wedge T_j(t_2) \right)$. Here, the goal is to establish alternatives for defining a function B_3 for the recognition of class 3. The first expression defines a tensor product multi-resolution Haar wavelet-based function; the second defines a Boolean function. ...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

Such a "generalized Boolean function" of the form $\bigvee_j \bigvee_i c_{i,j} (T_i(t_1) \wedge T_j(t_2))$ could possibly include real-valued truth variables t_1 and t_2 ($t_1, t_2 \in [0,1]$); real-valued truth functions T_1 and T_2; and real-valued coefficients $c_{i,j}$. In such a real-valued setting, one could interpret the real value of a truth variable as a "probability" – indicating in a "fuzzy" way whether a truth variable is "more likely true or more likely false." One would also have to properly define generalized OR ($\vee$) and AND ($\wedge$) algebraic operators for this real-valued scenario.



t_1	t_2	t_3	$\vee$	$\wedge$
0	0	0	f	F
1	0	0	t	F
0	1	0	t	F
1	1	0	t	F
0	0	1	t	F
1	0	1	t	F
0	1	1	t	F
1	1	1	t	T

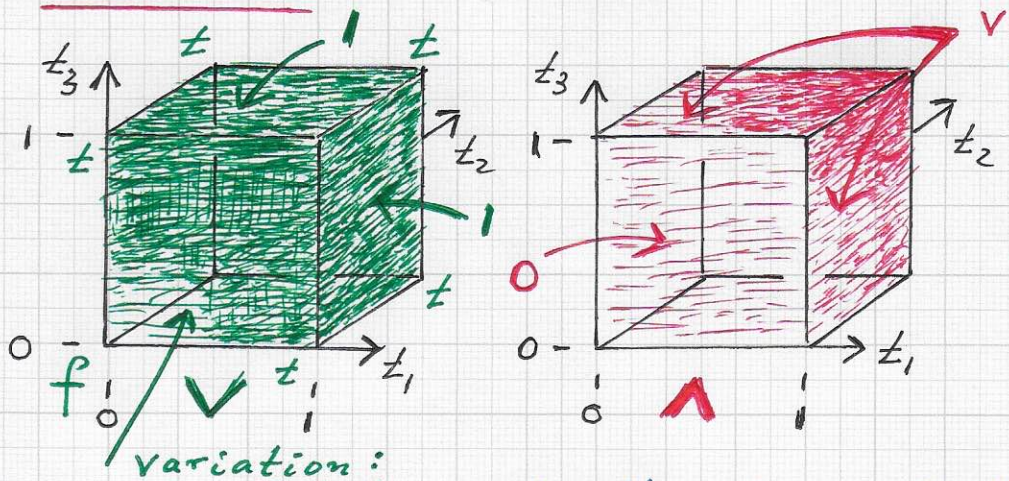
The left figure and table "mix" the binary truth values true (t, T) and false (f, F) and their associated integer values (0 ≙ FALSE, 1 ≙ TRUE), for the purpose of geometrically mapping truth variable triples (t_1, t_2, t_3) to the corners of the unit cube and visualizing the results of $\vee = t_1 \vee t_2 \vee t_3$ and $\wedge = t_1 \wedge t_2 \wedge t_3$ at the corners.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:...

In fact, instead of merely understanding the variables t_1, t_2 and t_3 in a binary way — where t_1, t_2, t_3 have only values in $\{0, 1\}$ — one should view them as real-valued variables — with values in $[0, 1]$ — representing "fuzzy Boolean variables" with real values.



The left figure visualize a variation of the resulting truth values of the generalized functions ∨ and ∧, as defined on the previous page. The gradual changes in the shading of the shown three faces of the unit cube indicate that the two functions are now defined over the entire $[0, 1]^3$ continuum domain, yielding real values. When using Haar wavelets, one must keep in mind that they can only approximate such a "smooth variation in a step-wise manner" — as they only piecewise constant.

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks:...

- **Note.** Approximation of Boolean functions (discrete, binary case).

b_1	b_2	b_3	B
F	F	F	T
T	F	F	F
F	T	F	T
T	T	F	T
F	F	T	F
T	F	T	F
F	T	T	T
T	T	T	T

The left table shows the results of a Boolean function B, depending on three truth variables b_1, b_2 and b_3. One exact, correct definition of B is

$$B = b_2 \vee (\neg b_1 \wedge \neg b_2 \wedge \neg b_3) .$$

An approximating, incorrect definition of a "good estimate" of B is the function B_{app}, e.g., the function

$$B_{app} = b_2 .$$

Thus, in addition to the use of K-map (Karnaugh-map) approaches for the minimization of (correct) representations of Boolean expressions, one can also consider the approximation of a Boolean function by eliminating specific terms from a correct representation of the function B by a simpler function B_{app} — that yields "in a small number of cases" an incorrect Boolean result, but calculated in a "significantly smaller amount of time." Whether one can determine and utilize such an approximation B_{app} depends on the requirements of a specific (data classification) application and a multitude of parameters.

- **Note.** Approximation of "generalized, real Boolean functions" via least squares and Haar wavelet expansions.

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

b_1	b_2	b_3	$\mathcal{B}$	
.05	.1	.1	.95	T
.1	.1	.1	.9	T
.9	.1	.05	.1	F
⋮	⋮	⋮	⋮	⋮
.8	.9	.1	.9	T
⋮	⋮	⋮	⋮	⋮
.1	.9	.95	.9	T
.9	.9	.95	.95	T

Considering a real-valued, non-binary function for determining "fuzzy Boolean values in the interval [0, 1]," for example, one could obtain the real b_i-values and the resulting real $\mathcal{B}$-value as shown in the left table. Here, $\mathcal{B} = \mathcal{B}(b_1, b_2, b_3)$, where $\mathcal{B} \leq 0.1$ implies FALSE (F) and $\mathcal{B} \geq 0.9$ implies TRUE (T), as shown next to the table. Generally, one is given a finite number of "similarity tuples" $(b_1^k, b_2^k, b_3^k) = \mathbb{b}_k$ with associated real, fuzzy Boolean values $\mathcal{B}_k$, $k = 1 \ldots s$, with s being the number of $\mathbb{b}_k$ and $\mathcal{B}_k$ values. One must understand $\mathcal{B}$ as a class-identifying function, e.g., $\mathcal{B} = \mathcal{B}_j$, used to recognize class-j materials/objects. The goal is the computation of a least-squares, best approximation for the s sample data using a multi-resolution Haar wavelet expansion — using a "meaningful number" of tensor product Haar wavelet functions. For example, the trivariate data in the shown table imply:

$$\sum_{i_3} \sum_{i_2} \sum_{i_1} c_{i_1,i_2,i_3} H_{i_1}(b_1^k) H_{i_2}(b_2^k) H_{i_3}(b_3^k) = \mathcal{B}_k \,, \quad k = 1 \ldots s.$$

Written more compactly, this linear system is defined as

$$\mathcal{B}(\mathbb{b}_k) = \sum_{\mathbb{i}} c_{\mathbb{i}} H_{i_1}(b_1^k) H_{i_2}(b_2^k) H_{i_3}(b_3^k) = \mathcal{B}_k \,, \quad k = 1 \ldots s.$$

The coefficient values are resulting from solving the over-determined system of the form $\mathbb{H}\mathbb{c} = \mathbb{B} \Rightarrow \mathbb{c} = (\mathbb{H}^T\mathbb{H})^{-1}\mathbb{H}^T\mathbb{B}$. ...

• Function $\mathcal{B}$: used to estimate "fuzzy Boolean values" for every $\mathbb{b}$-tuple; $\mathcal{B} = \mathcal{B}(\mathbb{b})$.