■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...

• **Signatures and discriminative power:**

1) Signatures with small Δ_{cl}-values have large discriminative power. When a tuple $(p_0', \ldots, p_c')$ has a small (large) distance value D_{cl} for such a "powerful" signature, then it is very probable that the unclassified tuple belongs (does not belong) to class cl.

2) Signatures with large Δ_{cl}-values have small discriminative power.

• **Note.** On page 21 (10/15/2022), the distance of a p_{cl}'-value on a specific P_{cl}-axis is more generally defined as a polynomial, i.e., $dist(P_{cl}) = (|P_{cl} - \bar{x}_{cl}| / \Delta_{cl})^n$, $n \in \{1, 2, 3, \ldots\}$. Thus, for the specific value $P_{cl} = p_{cl}'$ one obtains the distance $dist(p_{cl}') = (|p_{cl}' - \bar{x}_{cl}| / \Delta_{cl})^n$. Considering the specific example for a 2-class classification problem on page 9 (10/27/2022), the Δ_{cl}-values for the two signatures are

$$\Delta_0 = 1/8,\ \Delta_1 = 1/8,\ \Delta_2 = 1/8$$

for signature 1 and

$$\Delta_0 = 1/16,\ \Delta_1 = 1/16,\ \Delta_2 = 1/16$$

for signature 2. Using $n = 2$ (and weight $w_{cl} = 1$), one obtains

$$D_1 = 0^2 + (1/8 : 1/8)^2 + (1/8 : 1/8)^2 = 0 + 1 + 1 = 2,$$

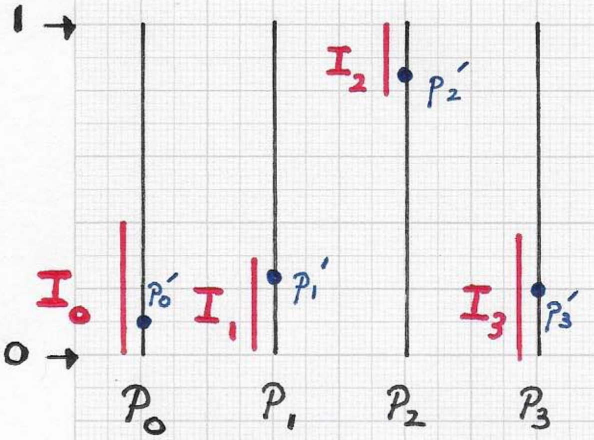
$$D_2 = (1/16 : 1/16)^2 + (11/16 : 1/16)^2 + (11/16 : 1/16)^2 = 1 + 121 + 121 = 243.$$

Thus, by using this quadratic polynomial the values for D_1 and D_2 are "much more different."

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:...



Exemplary case of a "clear" classification for class 2: The class-2 signature is highly discriminative, and $p_{cl} \in I_{cl}$, $cl = 0 \ldots 3$.

⇒ **Question:**

Given this class-2 signature, what is the "broadest allowable Boolean expression" involving I_{cl} and p_{cl}' that, when evaluated, should have the value TRUE (for class-2 classification)? How can one determine/learn this expression/rule?

• **Notes.** We use the example illustrated in the left figure to re-visit several of the important issues involved in the relationship between material class signatures, unclassified tuples $(p_0', \ldots, p_c')$, Boolean logic conditions and best-possible data classification.

i) The left figure shows the signature of class 2 and a tuple $(p_0', \ldots, p_3')$ that represents class 2. Why? The value of the expression

$$p_0' \in I_0 \wedge p_1' \in I_1 \wedge p_2' \in I_2 \wedge p_3' \in I_3$$

is TRUE (or 1). The following question arises: **Does a much larger set of Boolean expressions exist that, when evaluated for a/the tuple $(p_0', \ldots, p_c')$, should also yield TRUE?** Or: What is the (complete) set of Boolean expressions involving the operators AND, OR and NOT that – when TRUE – identify a tuple $(p_0', \ldots, p_c')$ as a class-cl tuple? **!**

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

● TP, TN, FP and FN – meaning in multi-class classification setting:

• TP – sample of class cl classified as a class-cl sample

• TN – sample not of class cl classified as a not-class-cl sample

• FP – sample of class different from cl classified as a class-cl sample

• FN – sample of class cl classified as a not-class-cl sample

⇒ Subject to the overarching performance requirements, the goal is to MAXIMIZE the values of TP_{no} and TN_{no} and MINIMIZE FP_{no} and FN_{no}.

ii) As part of the final classification step, one must map a real-valued p_{cl}' datum to a binary datum*, where $p_{cl}' \in [0,1]$ and * $b_{cl} \in \{0,1\}$. The mapping used for this purpose must have as its goal the optimization of the "final numbers" of true positives (TPs), true negatives (TNs), false positives (FPs) and false negatives (FNs) — used in the multi-class classification setting. In other words, one must determine and use the "optimal" mapping of the general form

$$b_{cl} = \begin{cases} 1, & \text{if CONDITION 1} \\ 0, & \text{otherwise} \end{cases}$$

CONDITION 1 would be a complex Boolean expression, most likely involving class-specific optimized threshold values. We count the numbers of TPs, TNs, FPs and FNs in a multi-class setting.

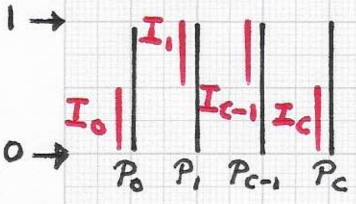
...

■ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks: ...

● The ideal, perfect binary patterns/signatures for a multi-class classification problem involving classes 0, 1, ..., C are the (C+1) bit strings (1, 0, ..., 0), ..., (0, ..., 0, 1).

⇒ "Good class signatures" should allow one to associate them clearly with exactly one of these (C+1) strings. Signatures MUST NOT exhibit behavior that would imply more than one bit with value 1.



Example of a "signature" that is not acceptable

⇒ Signatures must be "acceptable signatures" to support "acceptable classification."

iii) Specifically, one must solve a (combinatorial) optimization problem: Maximize the numbers of TPs and TNs, subject to $TP_{no} > \min TP_{no}$ and $TN_{no} > \min TN_{no}$, and minimize the numbers of FPs and FNs, subject to $FP_{no} < \max FP_{no}$ and $FN_{no} < \max FN_{no}$. In fact, an optimization procedure can terminate when the threshold conditions for TP_{no}, TN_{no}, FP_{no} and FN_{no} are satisfied, ASSUMING THAT THIS IS EVEN POSSIBLE.

iv) We must define this optimization problem for our multi-class classification setting, where classes 1, ..., C must be correctly recognized – and a "class 0" is a collective class representing all materials that do not belong to any of the classes 1, ..., C. • • •

▪ OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... **v)** First, we establish a methodology and terminology to characterize the simple case concerned only with two classes, 0 and 1, and the combinatorial possibilities when two objects must be classified. We use these abbreviations:

$T_0\,(T_1)$ = TRUE classification as class 0 (1)

$F_0\,(F_1)$ = FALSE classification as class 0 (1)

The following table summarizes the possibilities:

truth	classi-fication	type	number of T_0	T_1	F_0	F_1	ratio [%] of TRUE	: FALSE	
0 0	0 0	T_0 T_0	2	0	0	0	100	0	C
0 0	0 1	T_0 F_1	1	0	0	1	50	50	
0 0	1 0	F_1 T_0	1	0	0	1	50	50	
0 0	1 1	F_1 F_1	0	0	0	2	0	100	
0 1	0 0	T_0 F_0	1	0	1	0	50	50	
0 1	0 1	T_0 T_1	1	1	0	0	100	0	C
0 1	1 0	F_1 F_0	0	0	1	1	0	100	
0 1	1 1	F_1 T_1	0	1	0	1	50	50	
1 0	0 0	F_0 T_0	1	0	1	0	50	50	
1 0	0 1	F_0 F_1	0	0	1	1	0	100	
1 0	1 0	T_1 T_0	1	1	0	0	100	0	C
1 0	1 1	T_1 F_1	0	1	0	1	50	50	
1 1	0 0	F_0 F_0	0	0	2	0	0	100	
1 1	0 1	F_0 T_1	0	1	1	0	50	50	
1 1	1 0	T_1 F_0	0	1	1	0	50	50	
1 1	1 1	T_1 T_1	0	2	0	0	100	0	C

→ truth: the correct classes of the two objects

→ classification: the classification result for the two objects

→ type: the four possible classification result types T_0, T_1, F_0, F_1

→ number of...: the numbers of the four result types

→ ratio [%] of...: overall ratio of TRUE vs. FALSE classifications

The four correct classifications ("C") represent 4 of 16 combinatorially possible results.

...